UNIVERSITY OF ŽILINA
Faculty of Management Science
and Informatics

Cryptographic Systems

CCNA Security v2.0 / Network Security 1.0
Chapter 7 / Modules 15 - 17

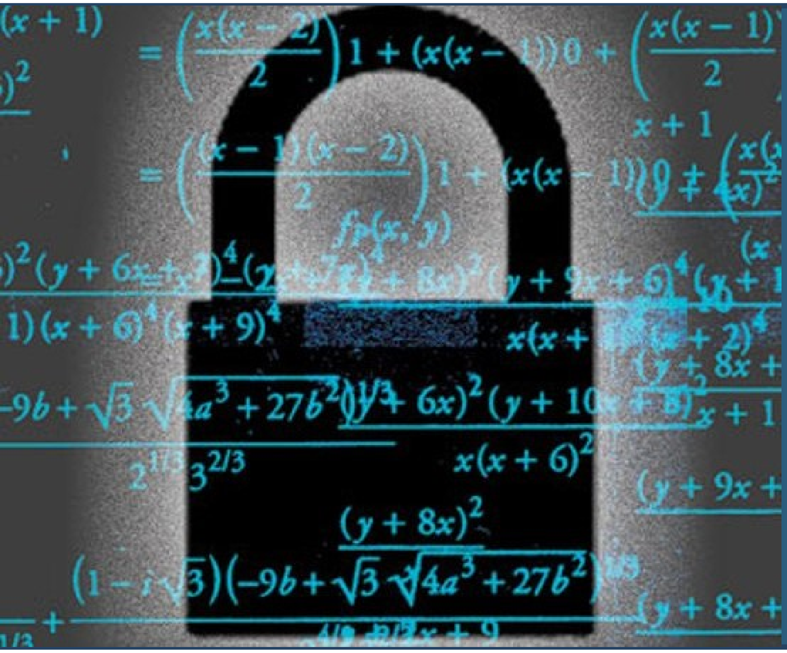
Bezpečnosť informačných sietí – KIS FRI UNIZA
Aktualizované v rámci projektu KEGA 026TUKE-4/2021.

* KATEDRA INFORMAČNÝCH SIETÍ * FAKULTA RIADENIA A INFORMATIKY ŽU

CISCO
Networking Academy

Chapter Outline

- Cryptographic Services
- Basic Integrity and Authenticity
- Confidentiality
- Public Key Cryptography



Cryptographic Services

Upon completion of this Section, you should be able to:

- Explain the requirements of secure communications including integrity, authentication, and confidentiality.
- Explain cryptography.
- Describe cryptoanalysis.
- Describe cryptology.

To secure a …

… network

- device hardening (routers, switches, servers, and hosts)
- authentication, authorization, and accounting (AAA)
- access control lists (ACLs)
- firewall / IPS
- Securing end points and services
  - AV
  - Email protecti0n, malware protection, web protection
- …

… communication/traffic

- Use of cryptographic methods
  - Cryptography: The development and use of codes
  - Cryptanalysis: breaking of codes
  - Cryptology: the science of making and breaking secret codes
- Three primary objectives of securing communication - CIA
  - **Confidentiality**
    - Message cannot be deciphered
      - Or at least make it very difficult ☺
  - **Integrity**
    - Message is not altered
  - **Authentication**
    - message is not a forgery and does actually come from whom it states

Authentication

- Message is not a fake (forgery) and does actually come from whom it states (i.e. prove a sender|
  - Usable for email or IP spoofing
- Two primary methods
  - Authentication services
    - Both sites share a secret/identity (for example PIN, AAA …)
  - Data nonrepudiation services
    - allows the sender of a message to be uniquely identified
      - And a sender cannot deny having been the source of that message (for example the check sign)
    - Rely on the fact that a sender has the unique characteristics or signatures for how that message is treated
- Mechanisms
  - hash message authentication code (HMAC) with keys
  - digital signature

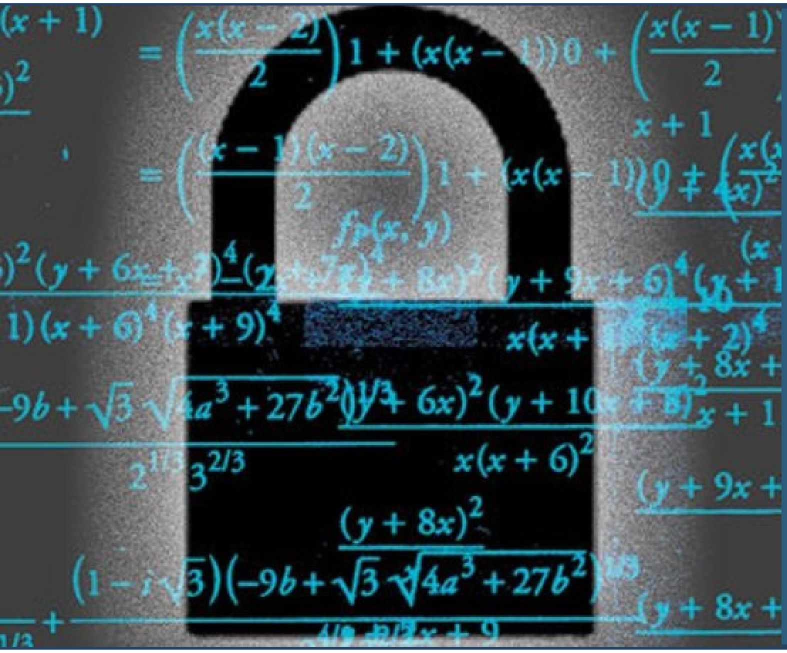
On behalf of the Client (autho
On behalf of the Developer (authorized sig

Data Integrity

- Ensures that the message is not altered (received message is identical to the sent one)
- Mechanisms => Hashing
  - MD5 or SHA hash-generating algorithms

Data Confidentiality

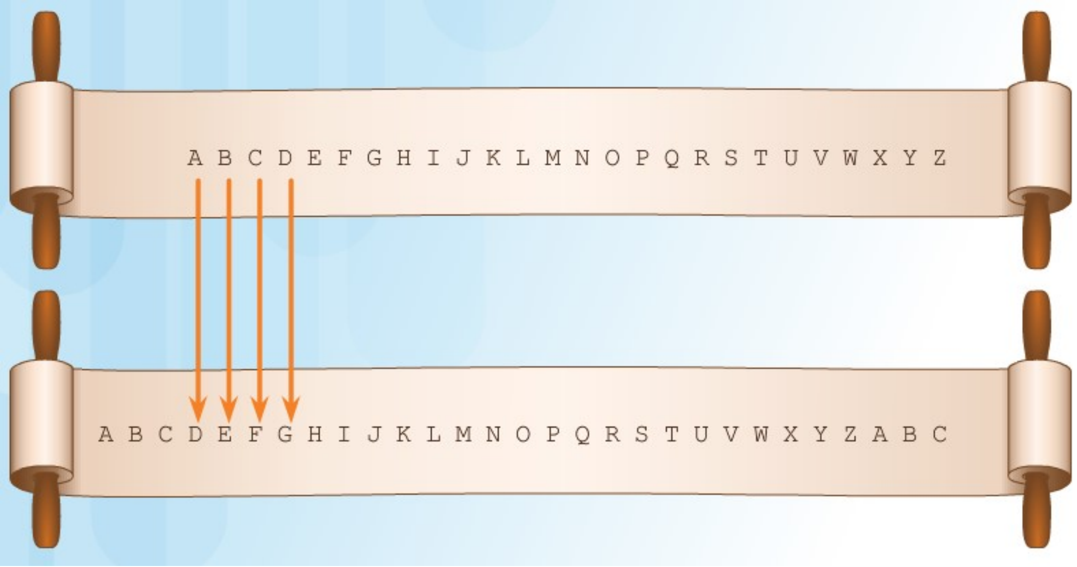
- Ensures privacy, i.e. message cannot be deciphered
  - And only a receiver can read it
- Mechanisms => Encryption (and hashing)
  - Symmetric encryption
    - Uses a shared key/keys (i.e. pre-shared key – PSK)
    - Data Encryption Standard (DES), 3DES (Triple DES), Advanced Encryption Standard (AES).
  - Asymmetric encryption
    - Does not use a previously shared password
    - Rivest, Shamir, and Adleman (RSA) and the public key infrastructure (PKI).
- Once enabled encryption we talk about
  - Readable data
    - Plain text, cleartext
  - Encrypted data
    - Encrypted text, ciphertext



Cryptography

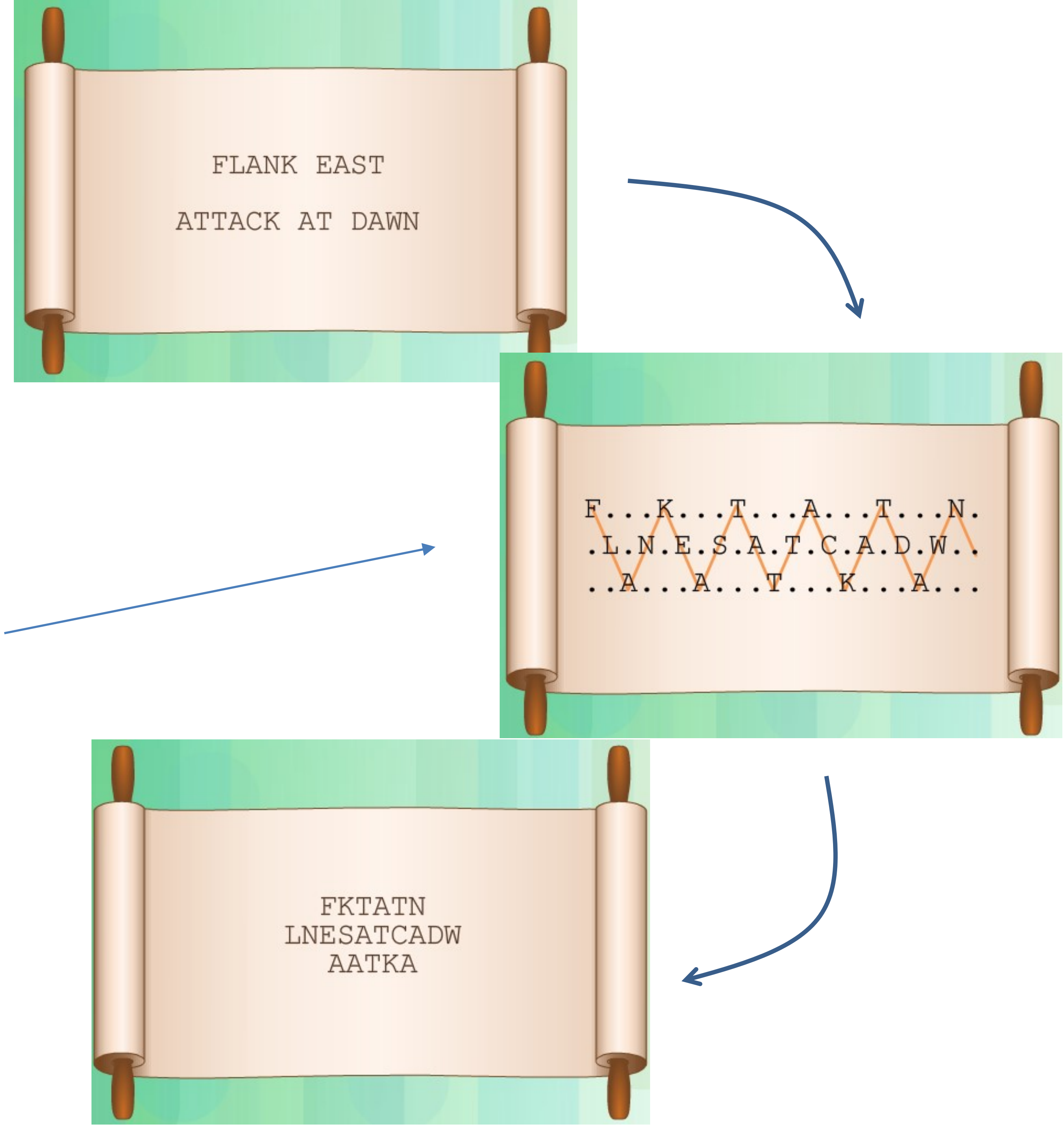
Cryptography - Creating Ciphertext

- Various cipher methods, physical devices, or aids have been used to encrypt and decrypt text
  - History: Scytale, Caesar Cipher, Enigma …
  - Each uses a specific algorithm, called a *cipher*
    - **Cipher** = series of well-defined steps that can be followed as a procedure when encrypting and decrypting messages
- Ciphertext can be creating using several methods:
  - Transposition
  - Substitution
  - One-time pad
  - …


A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Transposition Ciphers

- Basic cryptography operation
  - no letters are replaced
  - letters are simply rearranged
- Several transposition ciphers
  - Write text in reverse order
  - Rail Fence cipher (zigzag cipher)
    - Write letters downwards and diagonally on successive "rails" of an imaginary fence
    - Read message in rows
    - To decrypt, the number of rails have to be known
  - Columnar transposition
    - Rows of fixed length
  - ….
- DES/3DES also uses transposition



FLANK EAST

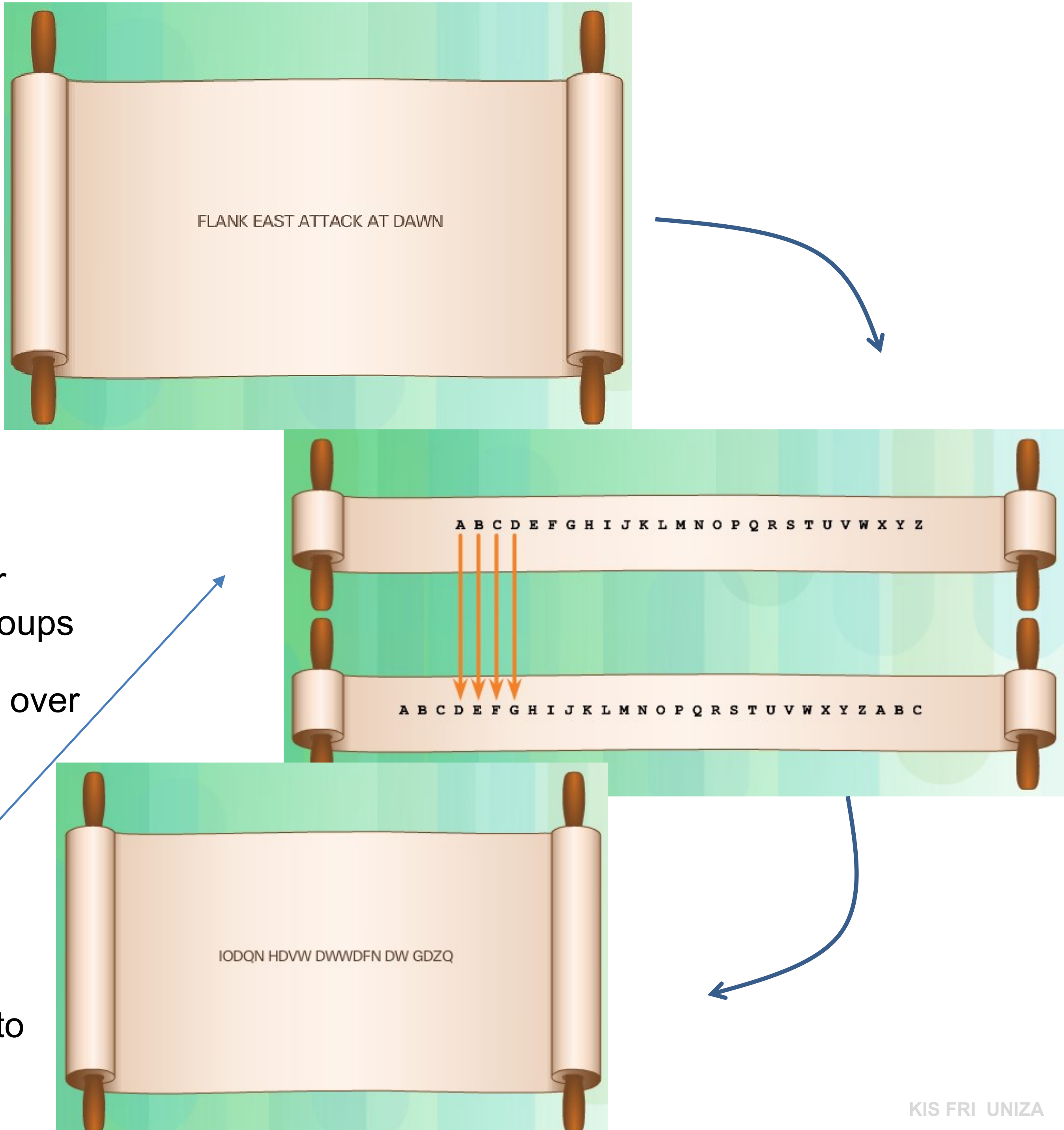
ATTACK AT DAWN

```
F...K...T...A...T...N.
.L.N.E.S.A.T.C.A.D.W..
..A...A...T...K...A...
```

FKTATN
LNESATCADW
AATKA

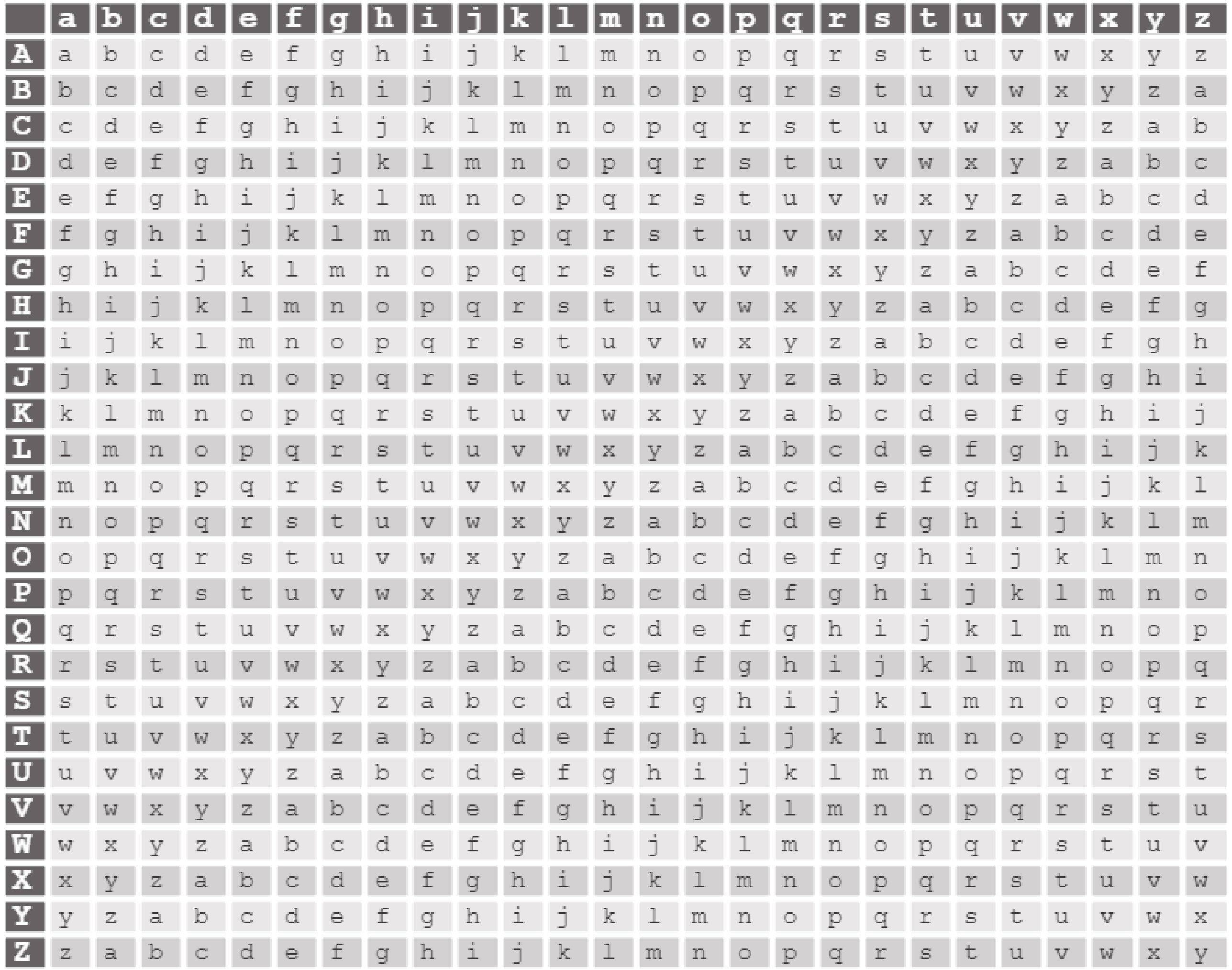
Substitution Ciphers

- Substitutes one letter of plaintext alphabet for another of encrypted alphabet
  - It may use one or more encrypted alphabets
- Ciphers
  - Number of different types
    - **Simple substitution**: one letter for another
    - **Polygraphic subst.:** operates on larger groups of letters
    - **Monoalphabetic cipher:** fixed substitution over the entire message
    - **Polyalphabetic cipher:** uses a number of substitutions at different positions in the message
    - …
  - Caesar cipher
    - simple, monoalphabetic substit
    - example uses key 3, shifts three positions to right


FLANK EAST ATTACK AT DAWN
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
IODQN HDVW DWWDFN DW GDZQ

Polyalphabetic substitution - Vigenère cipher

- 1585, unbreakable until 1863
- Uses table and secret key
- Encrypts text by using a different polyalphabetic key
- Key shift is identified using a shared key between sender and receiver
  - Letter of the message is encoded by intersection of the letter of the message taken from column and a row of identified by a letter of secret key



	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
A	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
B	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a
C	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b
D	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c
E	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d
F	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
G	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f
H	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g
I	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h
J	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i
K	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j
L	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k
M	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l
N	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m
O	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n
P	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
Q	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
R	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
S	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
T	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
U	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
V	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
W	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
X	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
Y	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
Z	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y

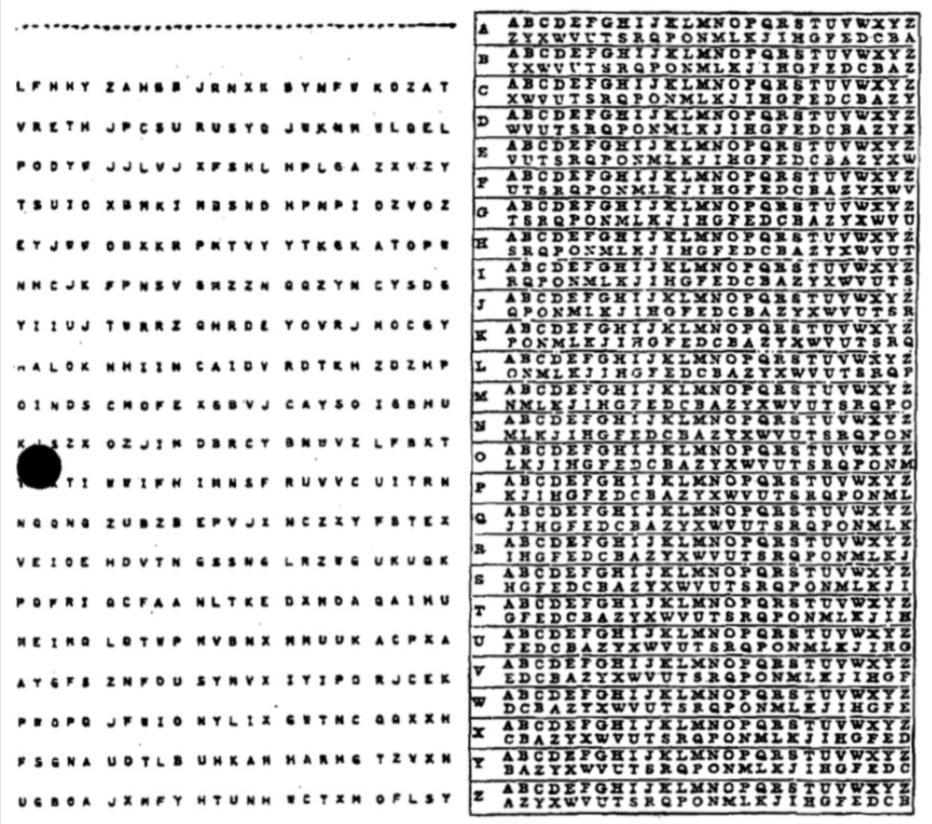
How the Vigenère Cipher Table works

- Suppose
  - sender and receiver have a shared secret key composed of these letters: SECRETKEY.
  - The sender uses this secret key to encode the plaintext FLANK EAST ATTACK AT DAWN:
    - The F (FLANK) is encoded by looking at the intersection of column F and the row starting with S (SECRETKEY), resulting in the cipher letter X.
    - The L (FLANK) is encoded by looking at the intersection of column L and the row starting with E (SECRETKEY), resulting in the cipher letter P.
    - The A (FLANK) is encoded by looking at the intersection of column A and the row starting with C (SECRETKEY), resulting in the cipher letter C.
    - The N (FLANK) is encoded by looking at the intersection of column N and the row starting with R (SECRETKEY), resulting in the cipher letter E.
    - The K (FLANK) is encoded by looking at the intersection of column K and the row starting with E (SECRETKEY), resulting in the cipher letter O.

- For example, SECRETKEYSECRETKEYSEC is required to encode FLANK EAST ATTACK AT DAWN:
  - Secret key: SECRETKEYSECRETKEYSEC
  - Plaintext: FLANKEASTATTACKATDAWN
  - Cipher text: XPCEOXKURSXVRGDKXBSAP

One-Time Pad (OTP) Ciphers

- Stream cipher co-invented by Gilbert Vernam (AT&T)
  - Uses a prepared key:
    - Consisting of an arbitrarily long, non-repeating sequence of bit/letters/character kept on paper tape
    - Each tape is used just once, i.e one time
    - Usually printed on flammable nitrocellulose
      - Used by KGB too
  - Encryption
    - Combines (pairs) bit/letter of the plaintext message with corresponding bits/letters of the key using modular addition (or XOR)
  - Decryption
    - Requires the same encryption key on a paper tape to reproduce plaintext
      - *Note: Cannot be cracked if using one-time and unique pre-shared key the same size as, or longer than, the message being sent [wiki]*
  - Example: RC4
- Problems
  - Generation of random data for keys
  - Can not use the key more times (easy decrypt)



```
LFHHY ZAHBB JRNXK BYNFV KOZAT
VRETH JPCSU RUSYQ JWKNN ELQEL
PODYW JJLVJ XFSKL HPLGA ZXVZY
TSUIO XBMKI MBSND HPNPI OZVOZ
EYJWW OBXKR PKTVY YTKGK ATOPW
NHCJK FPNSV BRZZN QQZYN CYSDG
YIIUJ TWRRZ QHRDE YOVRJ HOCGY
HALOK NHIIN CAIDV RDTKH ZDZHP
OINDS CHOFE XGBVJ CAYSO IGBHU
K[illegible]SZX OZJIN DBRCY BNUVZ LFBXT
[illegible]TI WWIFH IHNSF RUVVC UITRN
NQQNQ ZUBZB EPVJX NCZXY FBTEX
VEIOE HDVTN GSSNG LRZWG UKUQK
PQFRI QCFAA NLTKE DXHDA QAIHU
HEIHQ LQTWP HVBNX HHUUK ACPXA
AYGFS ZNFDU SYHVX IYIPD RJCEK
PWOPQ JFWIO NYLIX GWTNC QQXXH
FSGNA UDTLB UHKAH HARHG TZVXN
UGBOA JXHFY HTUNH WCTXH OFLSY
```

A	ABCDEFGHIJKLMNOPQRSTUVWXYZ ZYXWVUTSRQPONMLKJIHGFEDCBA
B	ABCDEFGHIJKLMNOPQRSTUVWXYZ YXWVUTSRQPONMLKJIHGFEDCBAZ
C	ABCDEFGHIJKLMNOPQRSTUVWXYZ XWVUTSRQPONMLKJIHGFEDCBAZY
D	ABCDEFGHIJKLMNOPQRSTUVWXYZ WVUTSRQPONMLKJIHGFEDCBAZYX
E	ABCDEFGHIJKLMNOPQRSTUVWXYZ VUTSRQPONMLKJIHGFEDCBAZYXW
F	ABCDEFGHIJKLMNOPQRSTUVWXYZ UTSRQPONMLKJIHGFEDCBAZYXWV
G	ABCDEFGHIJKLMNOPQRSTUVWXYZ TSRQPONMLKJIHGFEDCBAZYXWVU
H	ABCDEFGHIJKLMNOPQRSTUVWXYZ SRQPONMLKJIHGFEDCBAZYXWVUT
I	ABCDEFGHIJKLMNOPQRSTUVWXYZ RQPONMLKJIHGFEDCBAZYXWVUTS
J	ABCDEFGHIJKLMNOPQRSTUVWXYZ QPONMLKJIHGFEDCBAZYXWVUTSR
K	ABCDEFGHIJKLMNOPQRSTUVWXYZ PONMLKJIHGFEDCBAZYXWVUTSRQ
L	ABCDEFGHIJKLMNOPQRSTUVWXYZ ONMLKJIHGFEDCBAZYXWVUTSRQP
M	ABCDEFGHIJKLMNOPQRSTUVWXYZ NMLKJIHGFEDCBAZYXWVUTSRQPO
N	ABCDEFGHIJKLMNOPQRSTUVWXYZ MLKJIHGFEDCBAZYXWVUTSRQPON
O	ABCDEFGHIJKLMNOPQRSTUVWXYZ LKJIHGFEDCBAZYXWVUTSRQPONM
P	ABCDEFGHIJKLMNOPQRSTUVWXYZ KJIHGFEDCBAZYXWVUTSRQPONML
Q	ABCDEFGHIJKLMNOPQRSTUVWXYZ JIHGFEDCBAZYXWVUTSRQPONMLK
R	ABCDEFGHIJKLMNOPQRSTUVWXYZ IHGFEDCBAZYXWVUTSRQPONMLKJ
S	ABCDEFGHIJKLMNOPQRSTUVWXYZ HGFEDCBAZYXWVUTSRQPONMLKJI
T	ABCDEFGHIJKLMNOPQRSTUVWXYZ GFEDCBAZYXWVUTSRQPONMLKJIH
U	ABCDEFGHIJKLMNOPQRSTUVWXYZ FEDCBAZYXWVUTSRQPONMLKJIHG
V	ABCDEFGHIJKLMNOPQRSTUVWXYZ EDCBAZYXWVUTSRQPONMLKJIHGF
W	ABCDEFGHIJKLMNOPQRSTUVWXYZ DCBAZYXWVUTSRQPONMLKJIHGFE
X	ABCDEFGHIJKLMNOPQRSTUVWXYZ CBAZYXWVUTSRQPONMLKJIHGFED
Y	ABCDEFGHIJKLMNOPQRSTUVWXYZ BAZYXWVUTSRQPONMLKJIHGFEDC
Z	ABCDEFGHIJKLMNOPQRSTUVWXYZ AZYXWVUTSRQPONMLKJIHGFEDCB

Cryptoanalysis

Cracking Code

- Cryptoanalysis (Cracking Code):
  - the practice and study of determining the meaning of encrypted information, without access to the shared secret key
  - Exist as long as cryptography
    - Target: read the enemy messages (german enigma vs. british Government Code and Cypher School (GC&CS) at Bletchley Park)

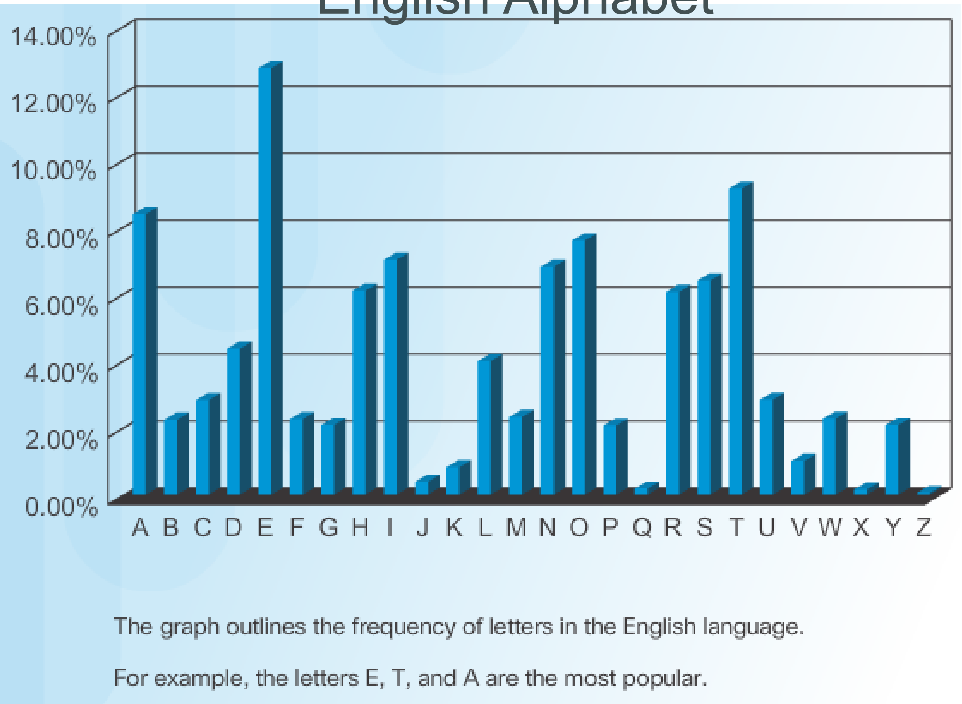
- Exists several cryptoanalysis methods
  - Brute-force method
    - Try every possible key (or combination) until one of them will work
  - Ciphertext method
    - Attacker has the ciphertext of several encrypted messages but no plaintexts
  - Known-Plaintext method
    - Attacker has the ciphertext of several messages and knows something about the plaintext
  - Chosen-Plaintext method
    - Attacker chooses which data the encryption device encrypts and observes the ciphertext output
  - Chosen-Ciphertext method
    - Attacker can choose different ciphertext to be decrypted and has access to the decrypted plaintext
  - Meet-in-the-Middle method
    - Attacker knows a portion of the plaintext and the corresponding ciphertext

Cryptoanalysis

Methods for Cracking Code

- Different approaches
  - Frequency analysis
    - Uses the specialties of given language,
    - For example
      - letters frequency and the use of popular characters in common language
        - E, T, and A are the most popular in English
        - J, Q, X, and Z are the least popular
      - the use of vowels (samohlásky)
      - …
    - In combination of other methods, for example brute force attack
      - Decryption of Caesar cipher

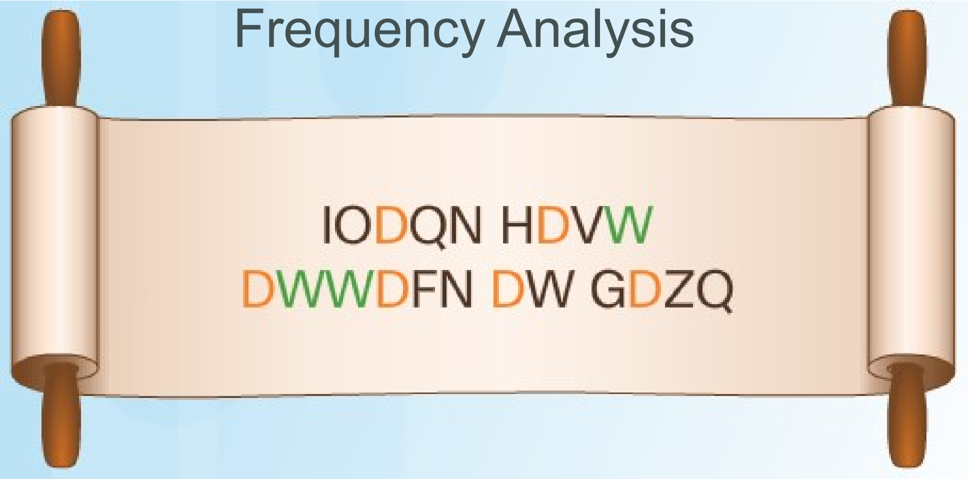
Frequency Analysis of the English Alphabet



The graph outlines the frequency of letters in the English language.

For example, the letters E, T, and A are the most popular.

Deciphering Using Frequency Analysis



IODQN HDVW
DWWDFN DW GDZQ

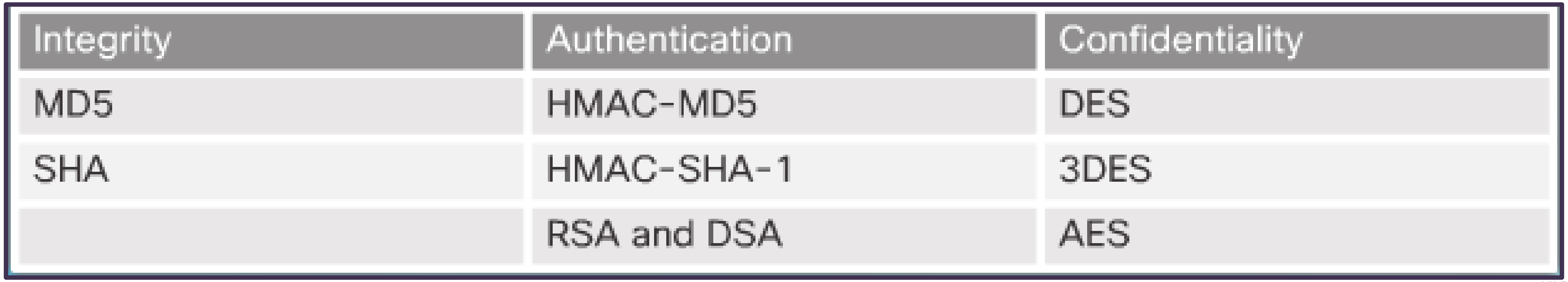
Cryptology

Making and Breaking Secret Codes

- **Cryptology = Cryptography + Cryptoanalysis**
  - The science of making and breaking secret codes
  - Two disciplines which each makes other stronger
    - Sometimes one is being ahead (WWII)
- **Cryptoanalysis**
  - Actually often used
    - by governments in military, diplomatic surveillance, national intelligence and defense services
    - by enterprises in testing the strength of security procedures,
    - by malicious hackers to get secrets
    - by mathematicians, scholars, and security forensic experts testing the strengths of ciphers and encryption methods
      - To prove vulnerability or resistance against known cryptanalytic attack

The Secret is in the Keys

- CIA features for communications and networking
  - Implemented in many ways using various protocols and algorithms
  - Choice depends and varies on required level of security or goals
    - i.e. MD5 faster but less secure to SHA
    - Symmetric less secure but faster as asymmetric
  - Actually using publicly available algorithm (comparing to secrecy of the historical algorithm like Enigma or Caesar)
    - where decryption requires knowledge of encryption keys
    - security therefore lies on secrecy of keys not the algorithm itself



Integrity	Authentication	Confidentiality
MD5	HMAC-MD5	DES
SHA	HMAC-SHA-1	3DES
	RSA and DSA	AES

Basic Integrity and Authenticity

Upon completion of the Section, you should be able to:

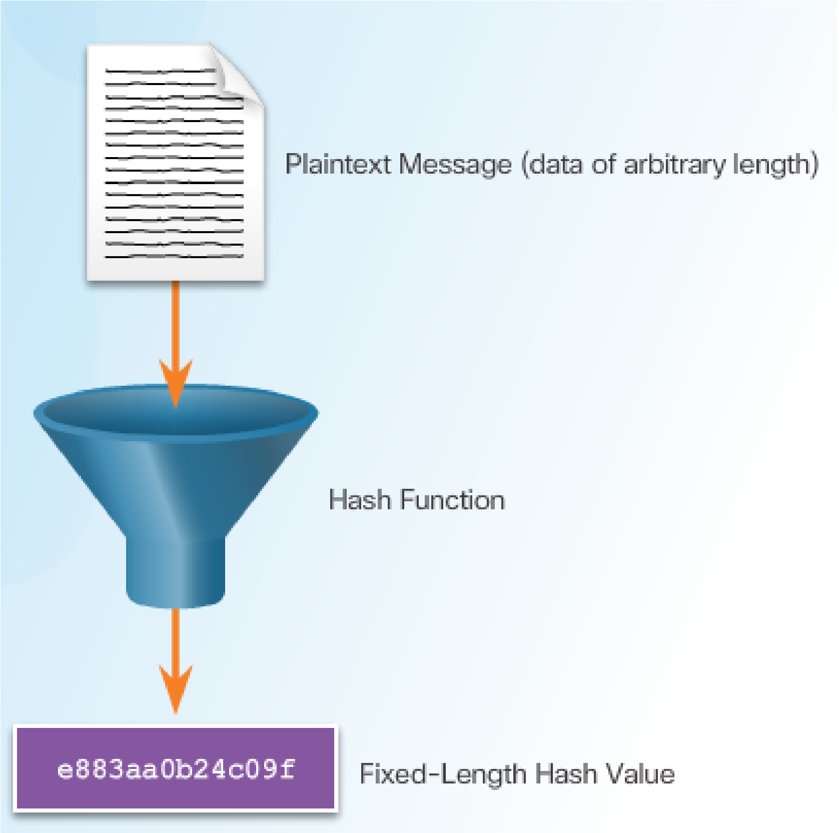
- Describe the purpose of cryptographic hashes.
- Explain how MD5 and SHA-1 are used to secure data communications.
- Describe authenticity with HMAC.
- Describe the components of key management.

Four elements of secure communications

- **Data Integrity**
  - Guarantees that the message was not altered. Any changes to data in transit will be detected.
- **Origin Authentication**
  - Guarantees that the message is not a forgery and does actually come from whom it states.
- **Data Confidentiality**
  - Guarantees that only authorized users can read the message. If the message is intercepted, it cannot be deciphered within a reasonable amount of time.
- **Data Non-Repudiation**
  - Guarantees that the sender cannot repudiate, or refute, the validity of a message sent. Nonrepudiation relies on the fact that only the sender has the unique characteristics or signature for how that message is treated.

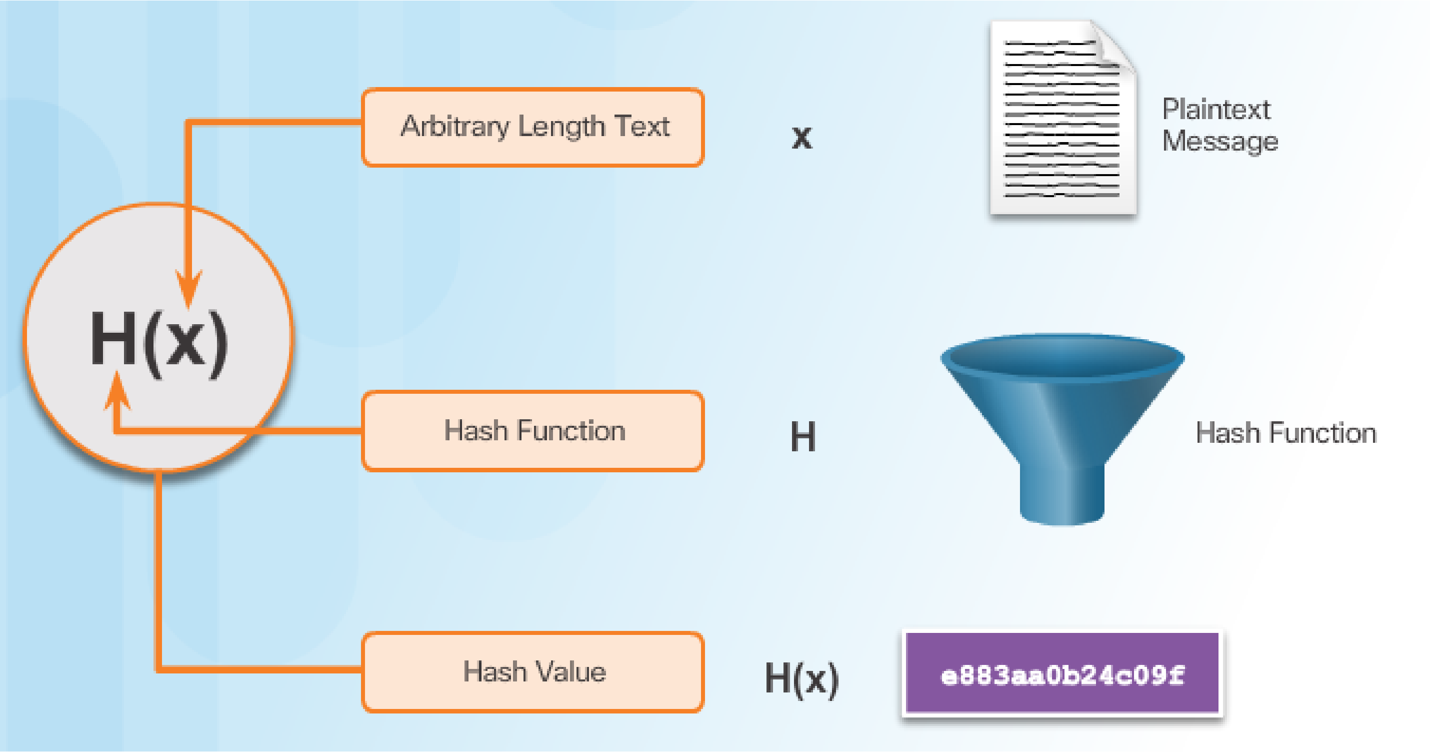
Cryptographic Hashes - Hash Function

- Used for integrity assurance
  - Sometimes the authenticity
- Based on an one-way mathematical function
  - relatively easy to compute
  - significantly harder to reverse
- Hash function
  - Takes a message (data)
  - Produce fixed-length, condensed representation = HASH or message digest
    - digital fingerprint
- Application examples
  - Message integrity
    - digitally signed contracts, and public key infrastructure (PKI) certificates, software packages
  - Proof of authenticity
    - when it is used with a symmetric secret authentication key (routing protocol auth, IPSec)
  - Authentication: PPP CHAP
  - And others

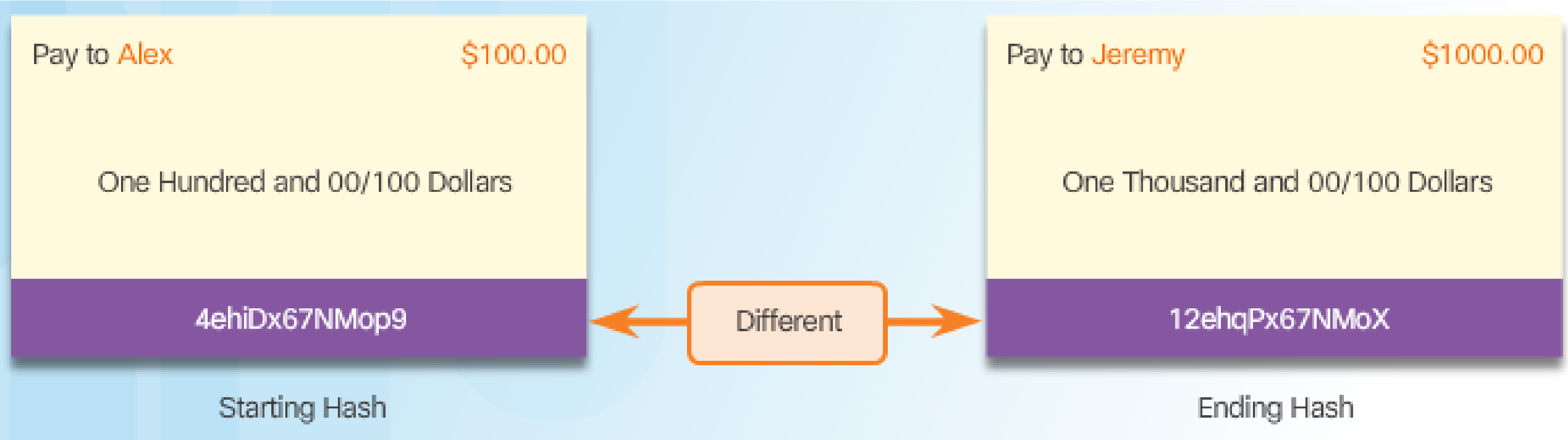


Cryptographic Hash Function Properties

- Math
  - $h = H(x)$
    - *H* takes an input *x* and returns a fixed-size string called the hash value *h*
- Properties
  - The input *x* can be any length
  - The output *h* has a fixed length
  - *H*(*x*) is relatively easy to compute for any given *x*
  - *H*(*x*) is one way and not reversible
    - hard to invert
  - *H*(*x*) is collision free
    - hard to find two different input values that result in the same hash value



Integrity - Well-Known Hash Functions



- Hash function
  - Helpful: Indicates accidental changes (results of error)
  - Attention: Cannot be used to guard against deliberate changes
    - There is no possibilities to indicate hash originality by hash function itself
    - Anyone may intercept the message, change the content and recalculate and append a new hash
      - MitM attacks
- Well known hash functions
  - MD5 with 128-bit digest
  - SHA-256 with 256-bit digest

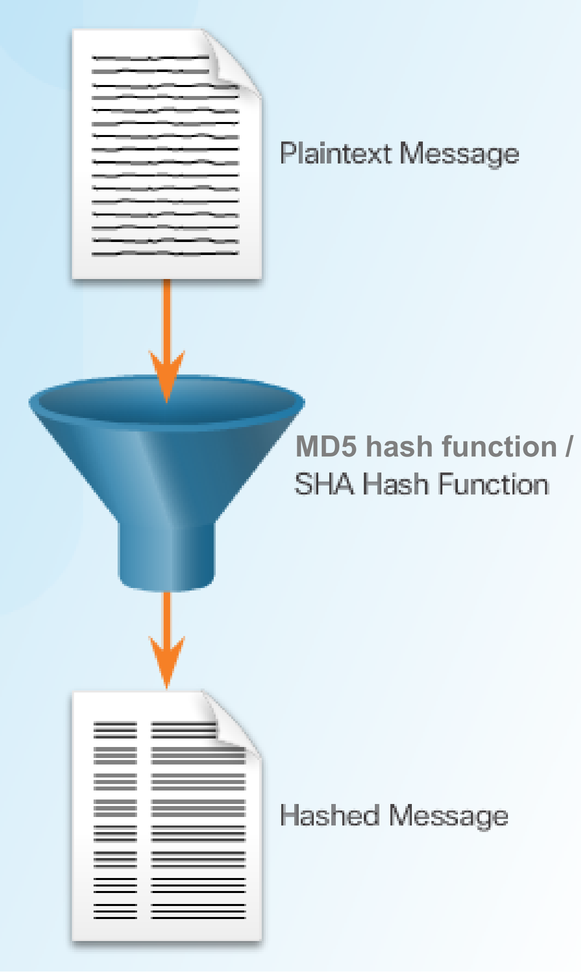
Message Digest 5 Algorithm / Secure Hash Algorithm

MD5

- Developed by Ron Rivest
- Used in a variety of Internet applications today
- Now considered as a legacy algorithm
  - The usage should be avoided
  - Use only when no better alternatives are available

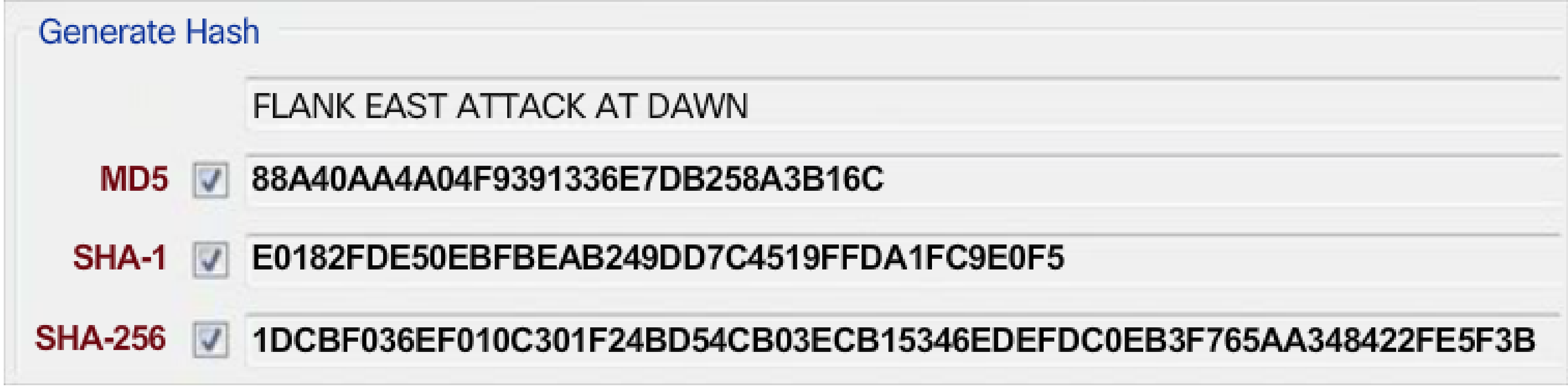
SHA

- Developed by The U.S. National Institute of Standards and Technology (NIST)
- Design is very similar to the MD5
  - But little bit slower
- Two versions
  - SHA-1
    - takes a message of less than 2^64 bits in length and produces a 160-bit message digest
    - Considered legacy, avoid use it
  - SHA-2 family
    - SHA-224 (224 bit)
    - SHA-256 (256 bit)
    - SHA-384 (384 bit)
    - SHA-512 (512 bit)



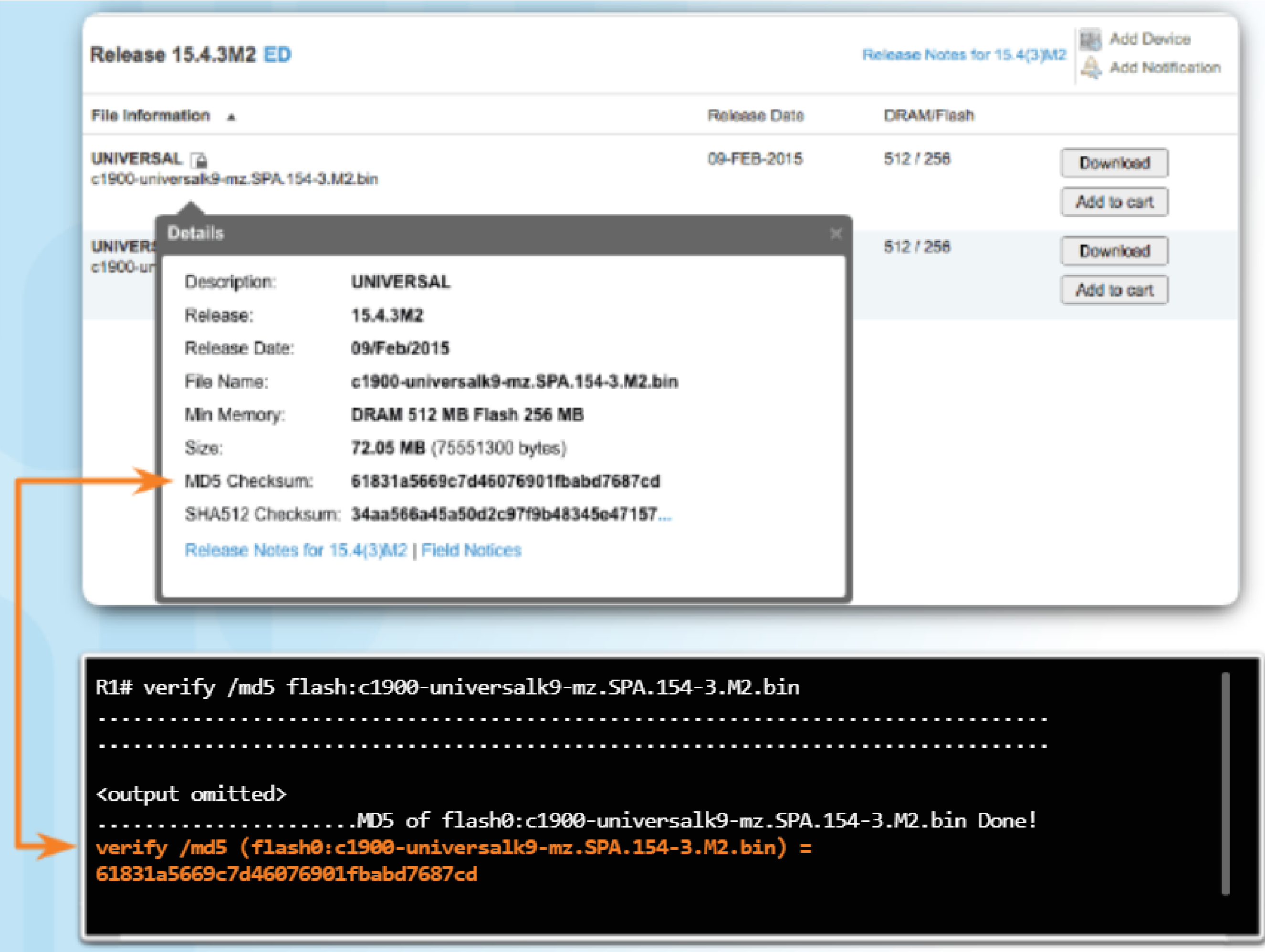
MD5 Versus SHA

- MD5, SHA-1
  - Faster,
  - but security flaws were discovered
  - Not recommended
- Good practise:
  - SHA-256 or higher
- Cisco signs IOS images
  - **`Verify /md5`**



Generate Hash

		FLANK EAST ATTACK AT DAWN
MD5	☑	88A40AA4A04F9391336E7DB258A3B16C
SHA-1	☑	E0182FDE50EBFBEAB249DD7C4519FFDA1FC9E0F5
SHA-256	☑	1DCBF036EF010C301F24BD54CB03ECB15346EDEFDC0EB3F765AA348422FE5F3B



Release 15.4.3M2 ED

File Information ▲	Release Date	DRAM/Flash	
UNIVERSAL c1900-universalk9-mz.SPA.154-3.M2.bin	09-FEB-2015	512 / 256	Download Add to cart
UNIVERS c1900-ur		512 / 256	Download Add to cart

Details

Description:	UNIVERSAL
Release:	15.4.3M2
Release Date:	09/Feb/2015
File Name:	c1900-universalk9-mz.SPA.154-3.M2.bin
Min Memory:	DRAM 512 MB Flash 256 MB
Size:	72.05 MB (75551300 bytes)
MD5 Checksum:	61831a5669c7d46076901fbabd7687cd
SHA512 Checksum:	34aa566a45a50d2c97f9b48345e47157...

Release Notes for 15.4(3)M2 | Field Notices

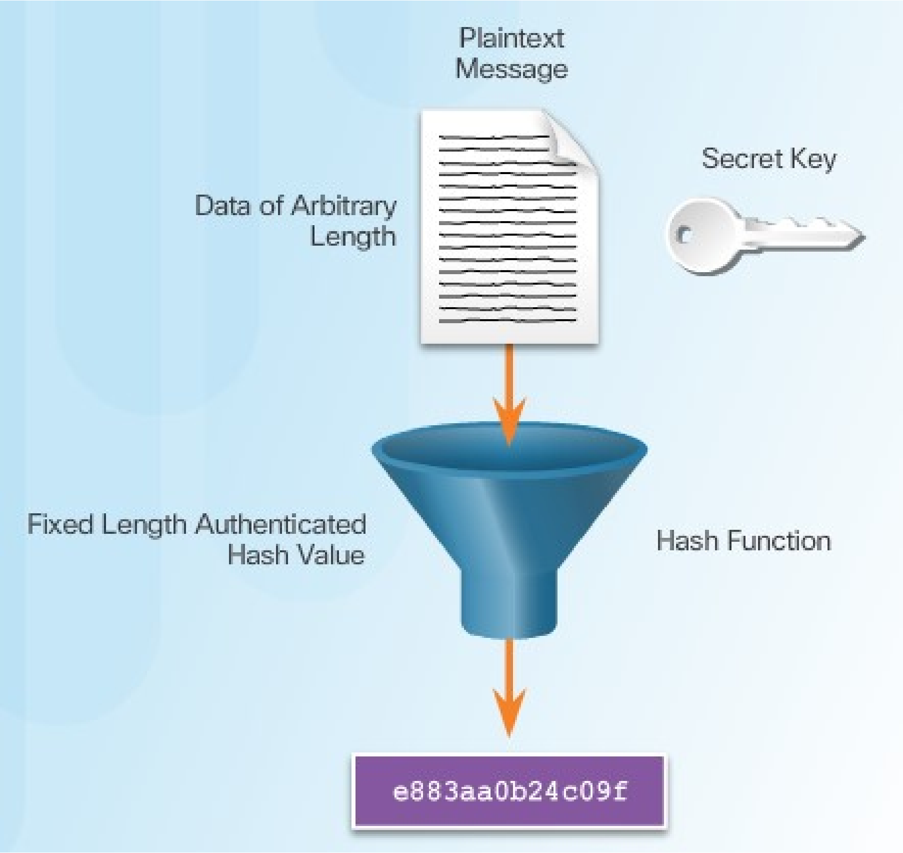
```
R1# verify /md5 flash:c1900-universalk9-mz.SPA.154-3.M2.bin
..............................................................................
..............................................................................

<output omitted>
......................MD5 of flash0:c1900-universalk9-mz.SPA.154-3.M2.bin Done!
verify /md5 (flash0:c1900-universalk9-mz.SPA.154-3.M2.bin) =
61831a5669c7d46076901fbabd7687cd
```

Authenticity with HMAC

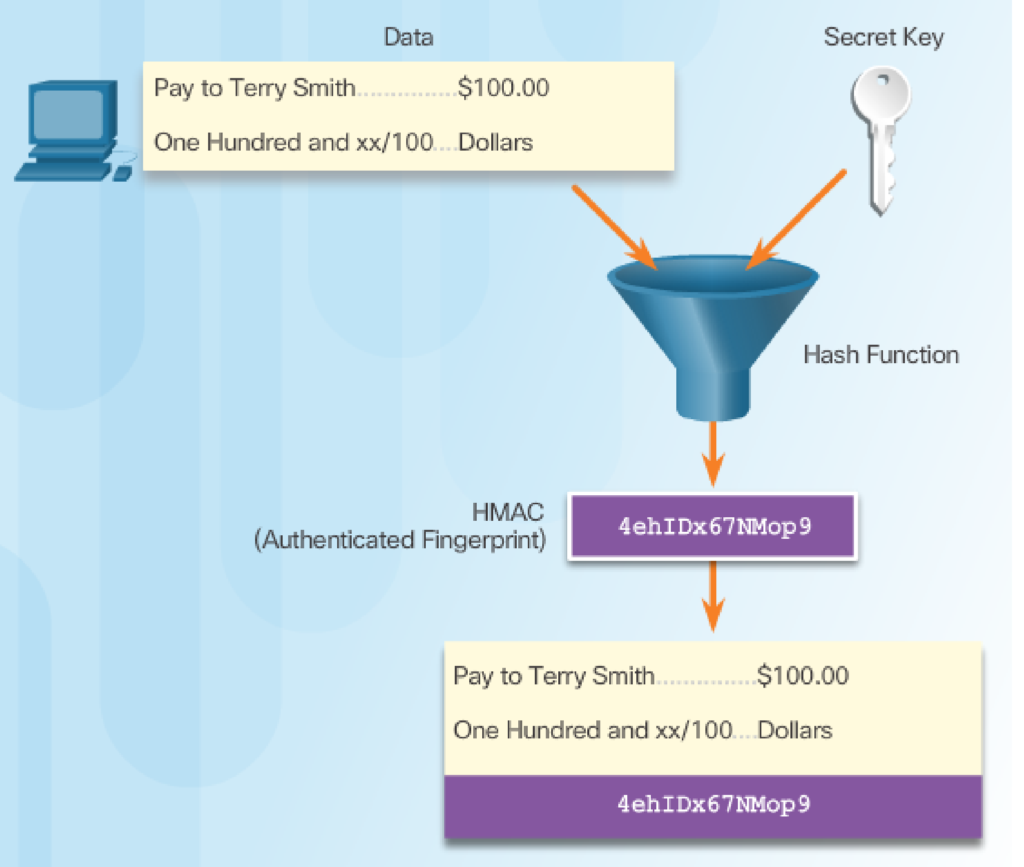
Keyed-Hash Message Authentication Code (HMAC or KHMAC)

- Types of message authentication code (MAC) functions
- Use an additional shared secret key as input to the hash function
  - Key has to be known to sender and receiver
    - Only the same inputs (message + key) produce the same Hash (digest)
  - Adds authentication to integrity assurance
    - Protects againt MiTM
- Two well-known HMAC functions
  - Keyed MD5 (HMAC-MD5)
  - Keyed SHA-1 (HMAC-SHA-1)
- Doubts?
  - How to distribute secret keys



HMAC Operation

Sender: Creating



Data

Secret Key

Pay to Terry Smith...............$100.00

One Hundred and xx/100....Dollars

Hash Function

HMAC
(Authenticated Fingerprint)

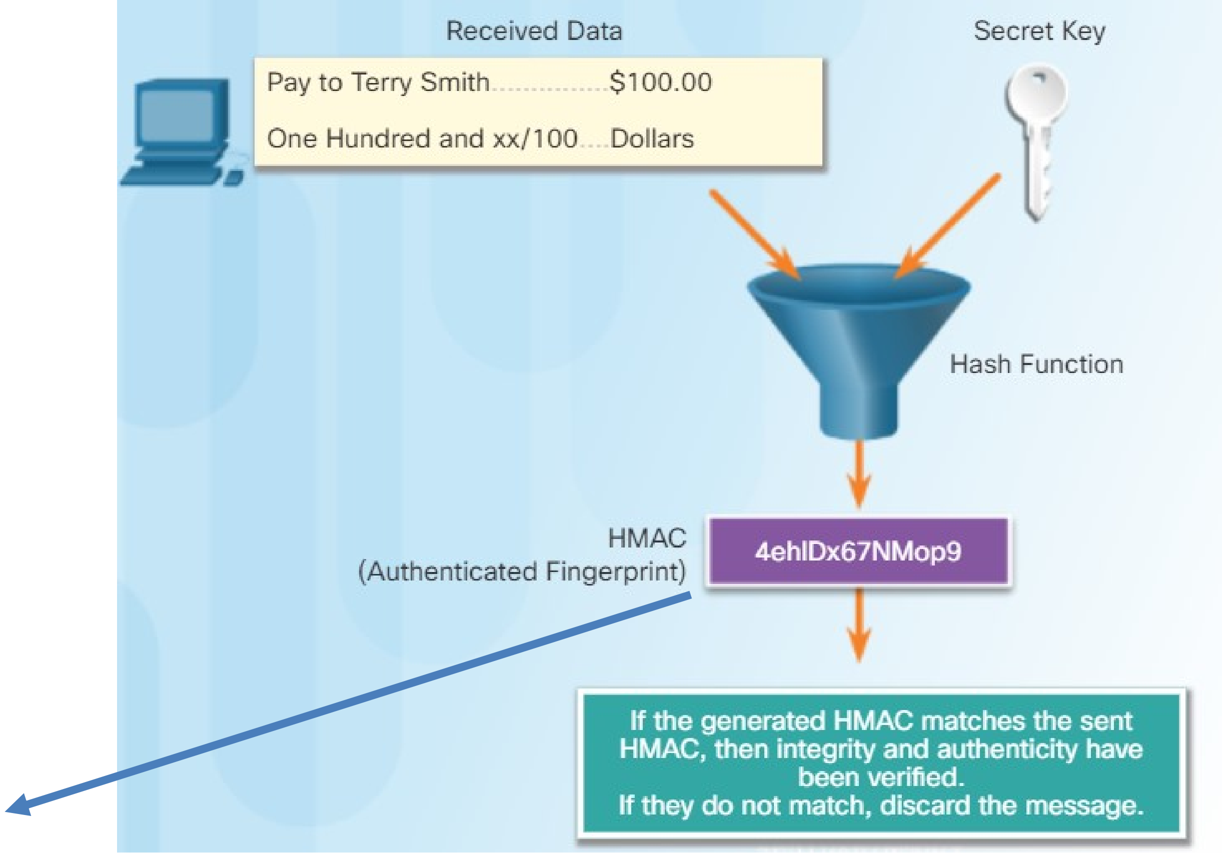
4ehIDx67NMop9

Pay to Terry Smith...............$100.00

One Hundred and xx/100....Dollars

4ehIDx67NMop9

Receiver: Verifying



Received Data

Secret Key

Pay to Terry Smith...............$100.00

One Hundred and xx/100....Dollars

Hash Function

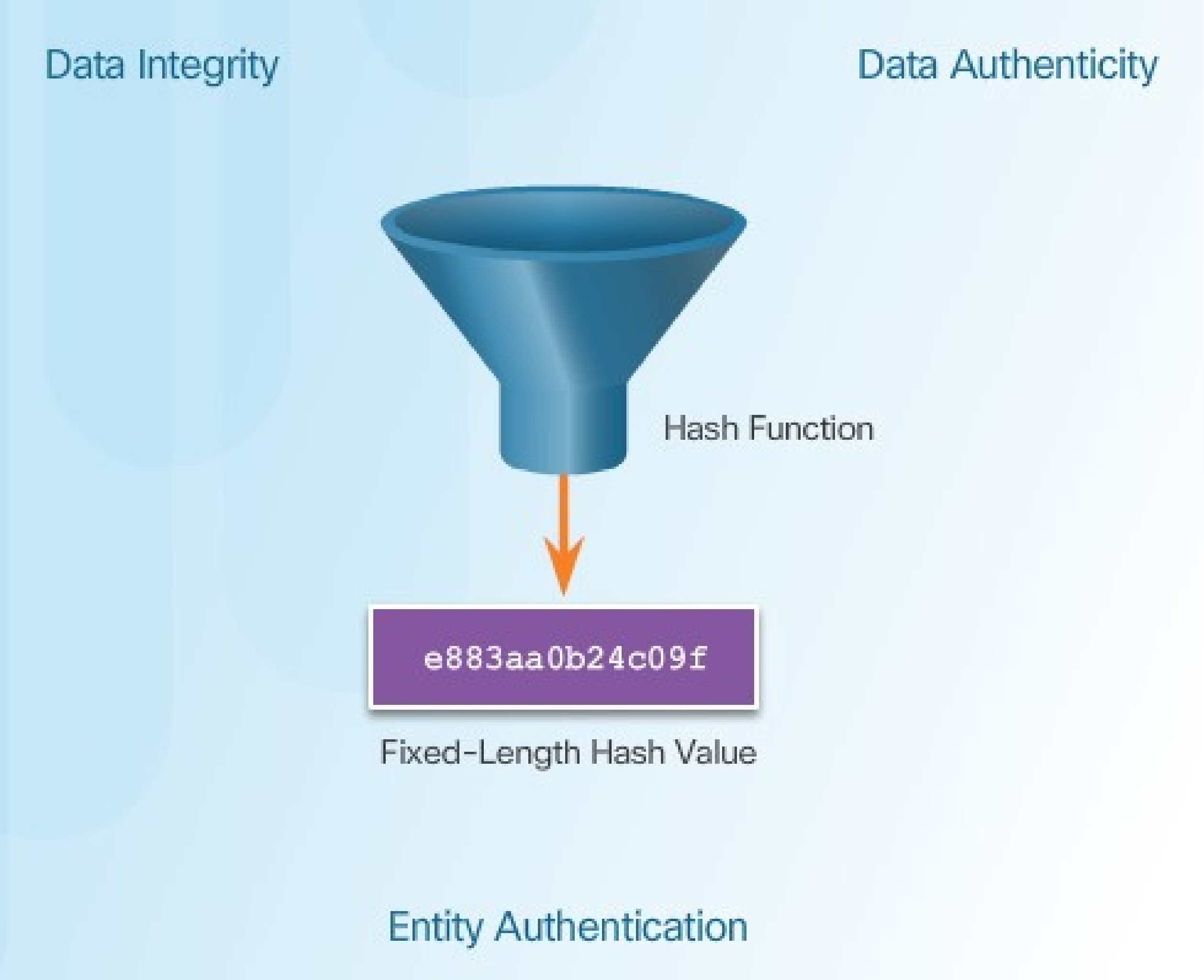
HMAC
(Authenticated Fingerprint)

4ehIDx67NMop9

If the generated HMAC matches the sent HMAC, then integrity and authenticity have been verified.
If they do not match, discard the message.

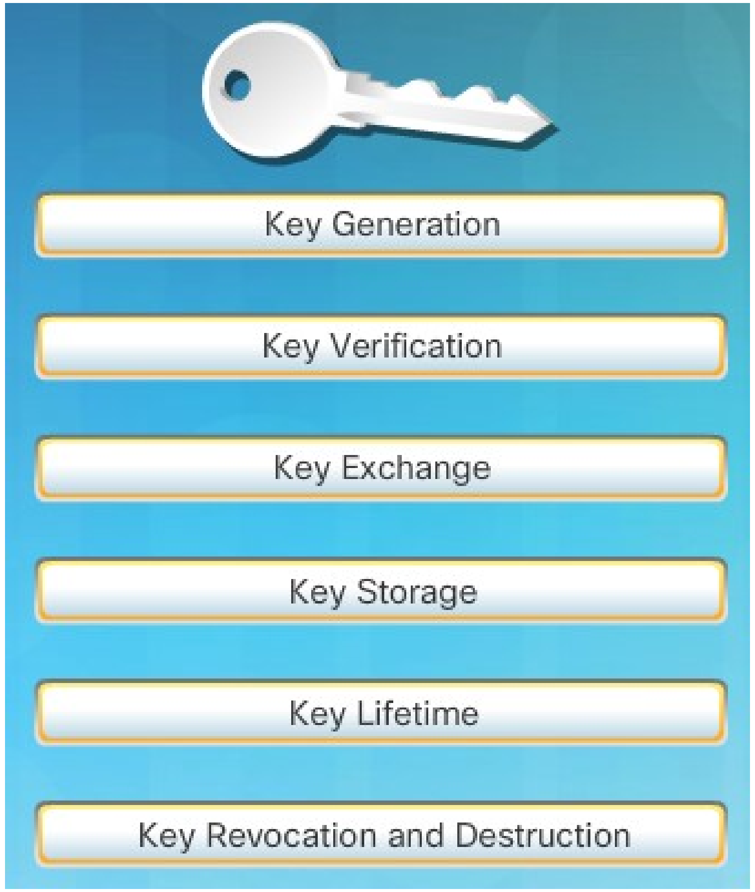
Hashing in Cisco Products

- Uses hashing for entity authentication, data integrity, and data authenticity
  - Router IOS
    - Authenticity of routing protocol updates
      - HMAC like MD5
      - RIPv2/ng, EIGRP, OSPF
  - IPSec gateways and clients
    - packet integrity and authenticity (MD5 and SHA-1)
  - Cisco IOS images verification



Key Management

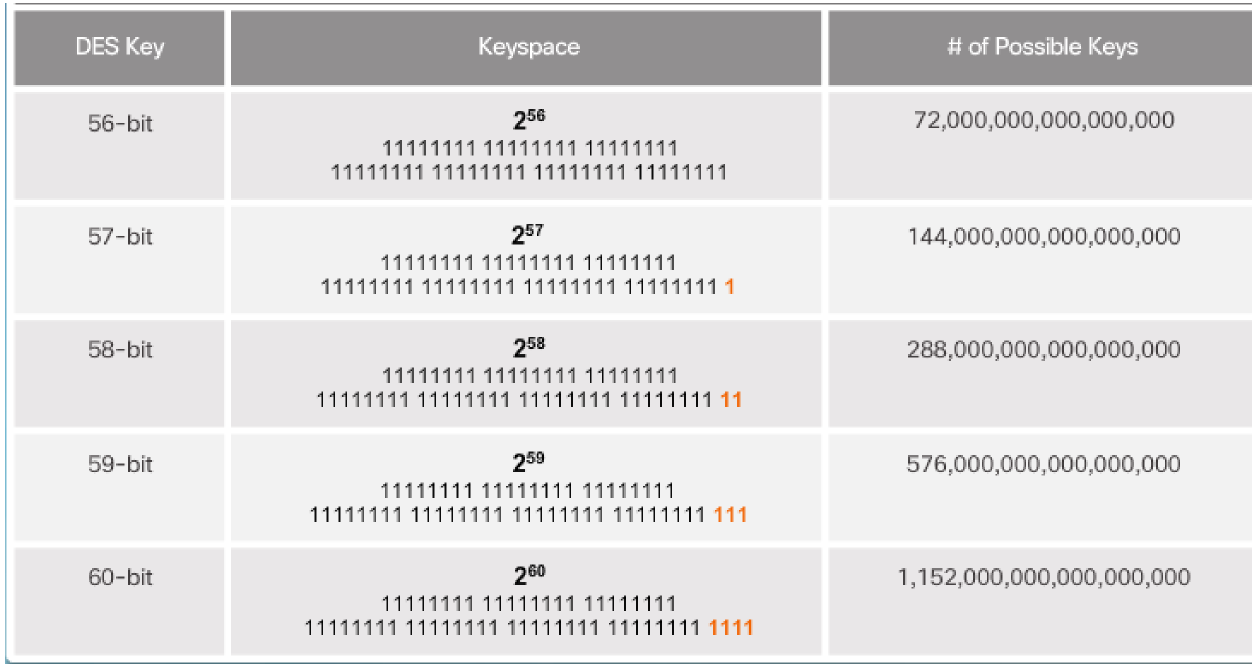
Characteristics of Key Management



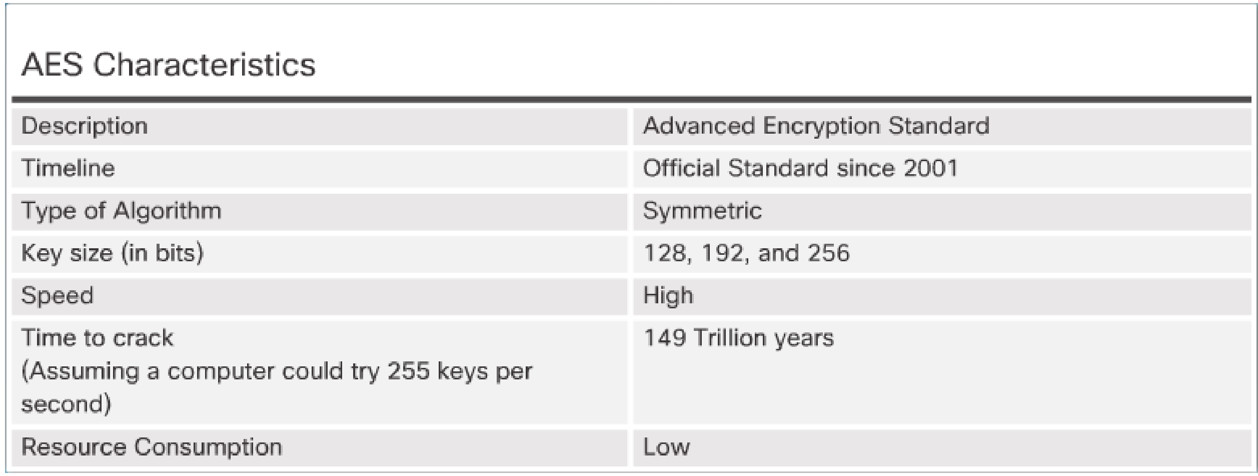
- Most difficult part of designing secure system
  - All modern cryptographic algorithms require key management procedures
  - Most attacks on cryptographic systems are aimed at the key management level
- Characteristics of key management
  - Key generation
    - By end users - communication parties - obsolete
    - Or automated (modern)
      - Good random generator is a need
  - Key verification
    - Procedures to identify weak keys and force re-generation
  - Key exchange
    - Procedure providing a secure key exchange mechanism over untrusted medium
  - Key storage
    - Help to avoid key leak (protect against memory leak, dumping and so on)
  - Key lifetime
    - Shorter is better
    - IPSec usualy has 24hours (long)
  - Key revocation and destruction
    - Indicates compromising of certain keys, starts deletion and recovery

Key Length and Keyspace

- Two terms describe keys
  - **Key length**
    - Size of the key measured in bits
  - **Keyspace**
    - Number of all possible key values (possibilities) generated for the key length
    - Increasing key length extend keyspace exponentially
- Almost every algorithm generates some weak keys
  - i.e. different key for which encryption is the same as decryption
  - DES for example
    - Alternating ones + zeros (0x0101010101010101)
    - Alternating 'F' + 'E' (0xFEFEFEFEFEFEFEFE)
    - '0xE0E0E0E0F1F1F1F1'
    - '0x1F1F1F1F0E0E0E0E’



DES Key	Keyspace	# of Possible Keys
56-bit	2^{56} 11111111 11111111 11111111 11111111 11111111 11111111 11111111	72,000,000,000,000,000
57-bit	2^{57} 11111111 11111111 11111111 11111111 11111111 11111111 11111111 1	144,000,000,000,000,000
58-bit	2^{58} 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11	288,000,000,000,000,000
59-bit	2^{59} 11111111 11111111 11111111 11111111 11111111 11111111 11111111 111	576,000,000,000,000,000
60-bit	2^{60} 11111111 11111111 11111111 11111111 11111111 11111111 11111111 1111	1,152,000,000,000,000,000



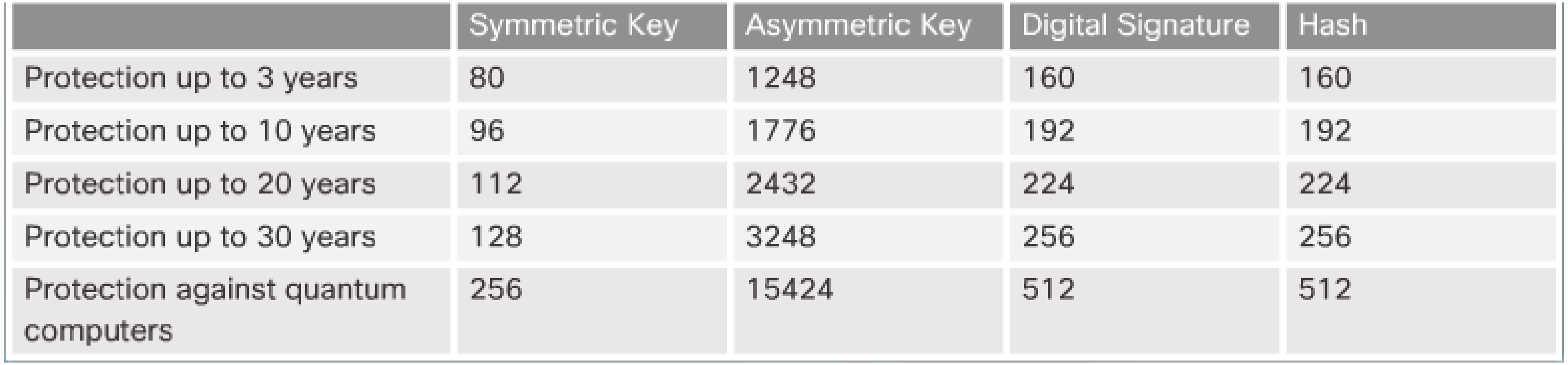
AES Characteristics

Description	Advanced Encryption Standard
Timeline	Official Standard since 2001
Type of Algorithm	Symmetric
Key size (in bits)	128, 192, and 256
Speed	High
Time to crack (Assuming a computer could try 255 keys per second)	149 Trillion years
Resource Consumption	Low

Types of Cryptographic Keys

- Types of cryptographic keys:
  - Symmetric keys
  - Asymmetric keys
  - Digital signatures
  - Hash keys

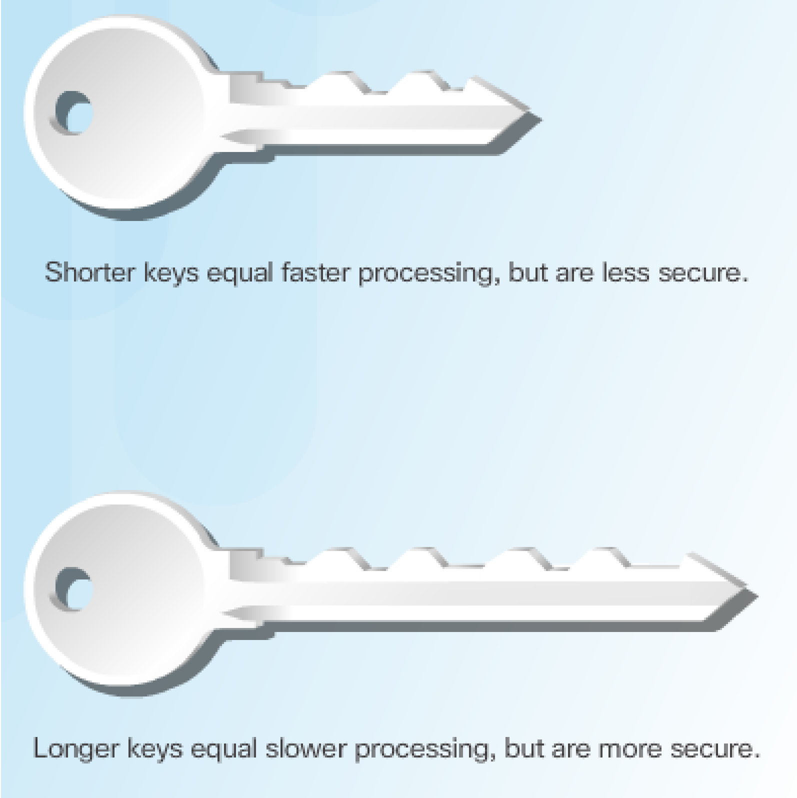
- Keys share some issues
  - Suitable key length
    - Should have keyspace big enough to make attack time-consuming
      - Protection against brute force
      - Take into a count computer power
    - More sensitive data should use longer keys

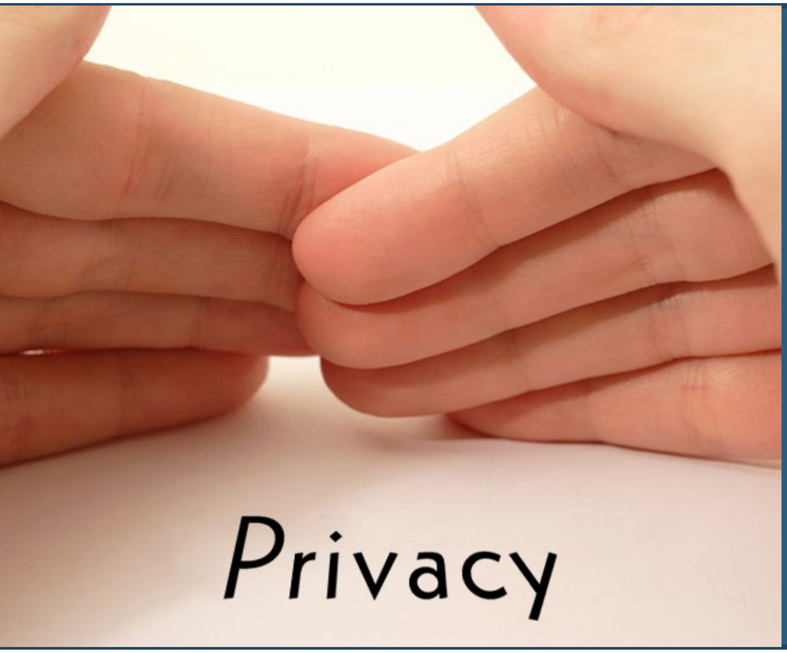


	Symmetric Key	Asymmetric Key	Digital Signature	Hash
Protection up to 3 years	80	1248	160	160
Protection up to 10 years	96	1776	192	192
Protection up to 20 years	112	2432	224	224
Protection up to 30 years	128	3248	256	256
Protection against quantum computers	256	15424	512	512

Choosing Cryptographic Keys

- But how big is enough?
  - Longer key = better
    - Always true
- Administrator should find some balance
  - Performance vs protection
    - Execution time
      - Key length influences algorithm execution
        - RSA runs slowly with large keys
    - Estimation of the attacker funding
      - Required time, processing resources, …
      - Assessing the risk of breaking
        - Estimating value information


Shorter keys equal faster processing, but are less secure.
Longer keys equal slower processing, but are more secure.



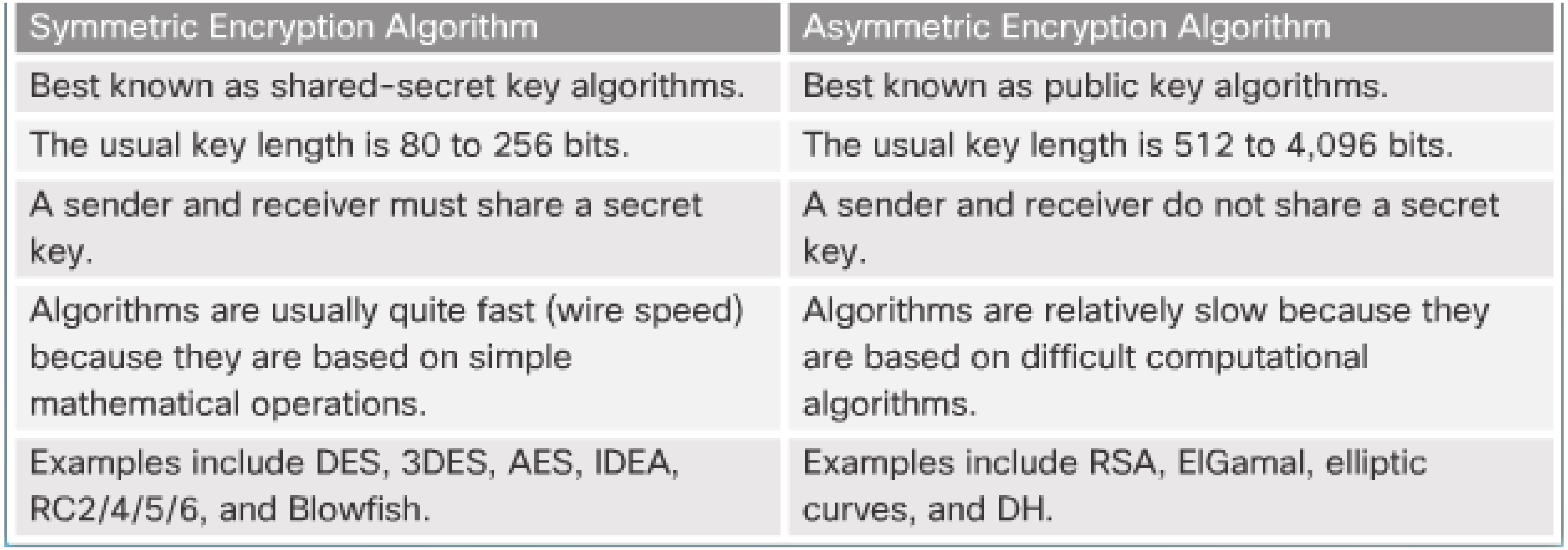
Confidentiality through encryption

Upon completion of the Section, you should be able to:

- Explain how encryption algorithms provide confidentiality.
- Explain the function of the DES, 3DES, and the AES algorithms .
- Describe the function of the Software Encrypted Algorithm (SEAL) and the Rivest ciphers (RC) algorithms.

Classes of Encryption Algorithms

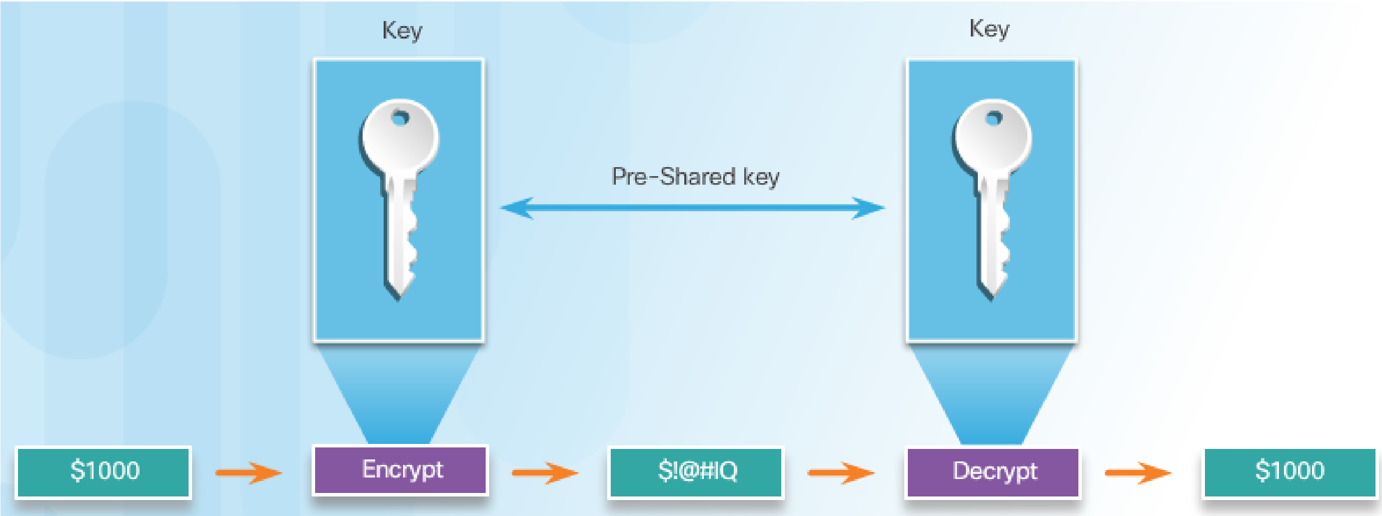
- Two approaches to ensure security using encryption
  - Protect algorithm
    - Secrecy of the algorithm itself; reveal of algorithm ==> need to change algorithm
  - Protect keys
    - Algorithms are publicly available and well known

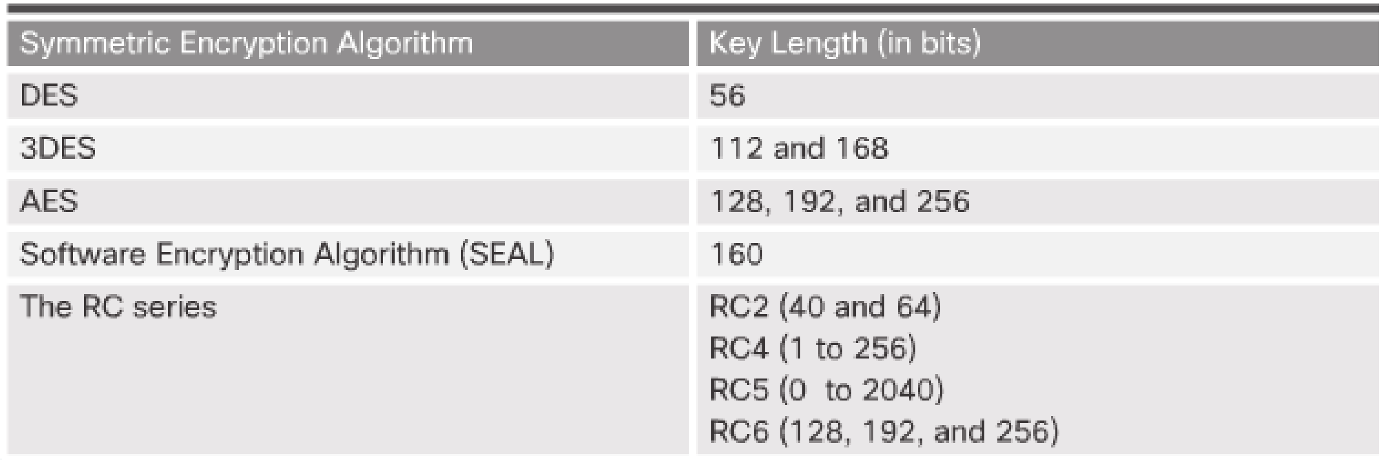


Symmetric Encryption Algorithm	Asymmetric Encryption Algorithm
Best known as shared-secret key algorithms.	Best known as public key algorithms.
The usual key length is 80 to 256 bits.	The usual key length is 512 to 4,096 bits.
A sender and receiver must share a secret key.	A sender and receiver do not share a secret key.
Algorithms are usually quite fast (wire speed) because they are based on simple mathematical operations.	Algorithms are relatively slow because they are based on difficult computational algorithms.
Examples include DES, 3DES, AES, IDEA, RC2/4/5/6, and Blowfish.	Examples include RSA, ElGamal, elliptic curves, and DH.

Symmetric Encryption

- Or secret key encryption
  - Encryption and decryption **use the same key**
- Popular and commonly used
  - Fast and speedy
    - Simpler math operations
    - Shorter keys
    - Easily accelerated in HW
    - Almost wire speed encryption
      - Useful for VPN and VoIP for example
  - Exists many of symmetric encryption algorithms
- Problems
  - Key distribution
    - Sender and the receiver must exchange keys somehow
    - Ideally using secured channel
  - Security of keys is the point

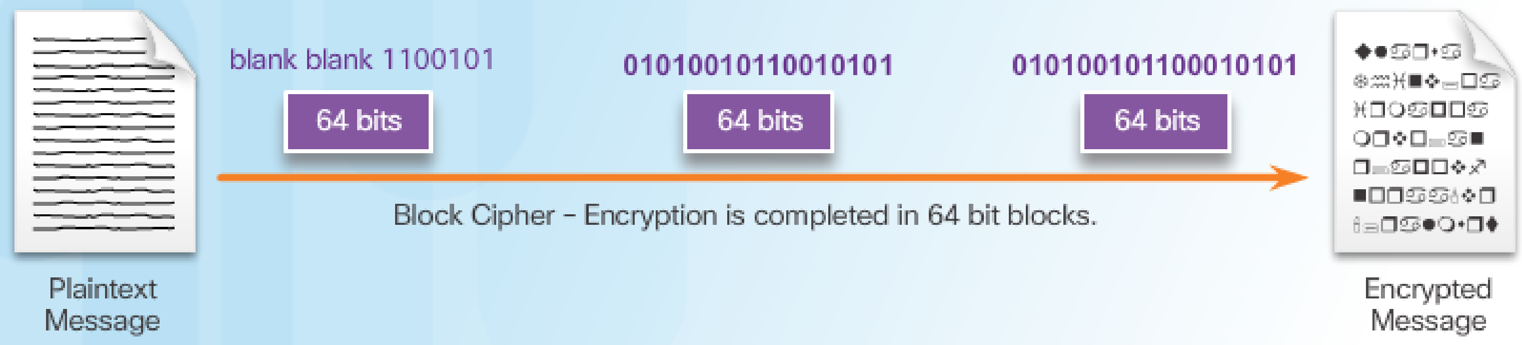




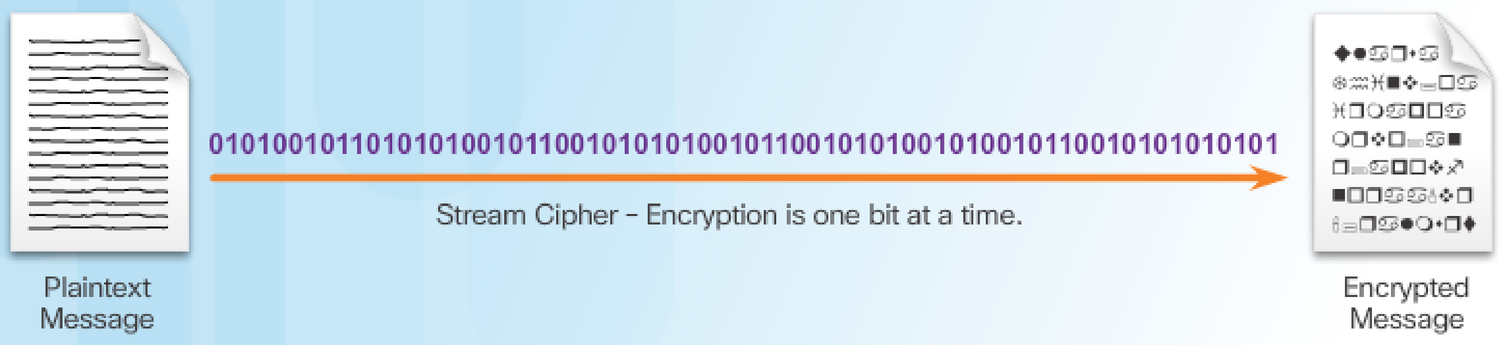
Symmetric Encryption Algorithm	Key Length (in bits)
DES	56
3DES	112 and 168
AES	128, 192, and 256
Software Encryption Algorithm (SEAL)	160
The RC series	RC2 (40 and 64) RC4 (1 to 256) RC5 (0 to 2040) RC6 (128, 192, and 256)

Symmetric Block Ciphers and Stream Ciphers

- Are the most commonly used techniques of symmetric algorithm
- **Block cipher**
  - Transforms a fixed-length block of plaintext (chunks) into a common block of ciphertext
    - 64 or 128 bits
  - Block is encrypted at one time
  - Increase data size
    - Output is bigger as input (multiples of block size)
- **Stream Cipher**
  - Encrypts message per bit/byte
    - *„Block cipher with a block size of 1 bit“*



Block Cipher - Encryption is completed in 64 bit blocks.

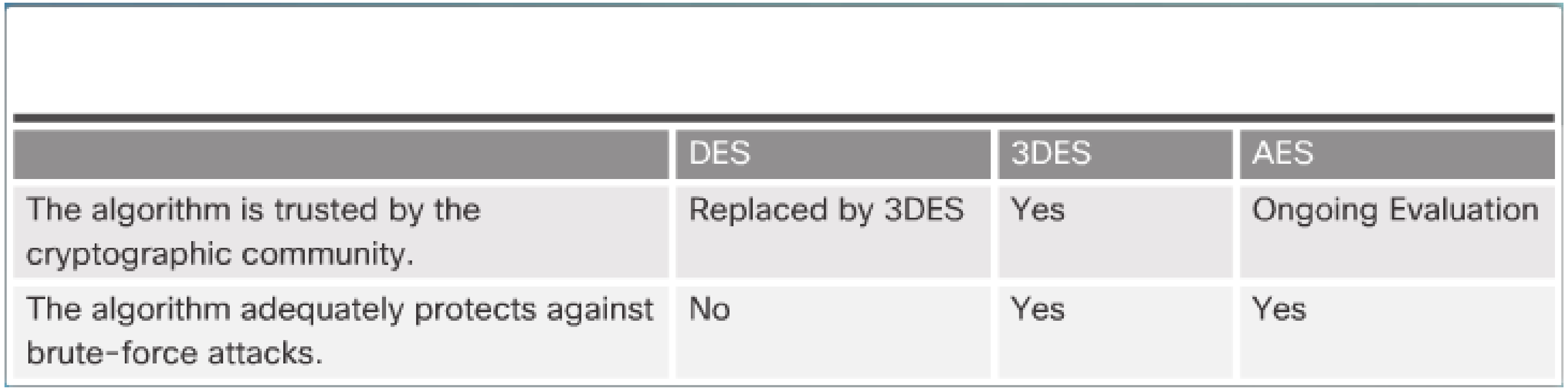


Stream Cipher - Encryption is one bit at a time.

- Much faster then block ciphers
- Do not increase output size
  - Example is RC4, DES, … Vigenère cipher
- Periodic
  - the key is of finite length
  - Then is repeating (if message is bigger)

Choosing an Encryption Algorithm

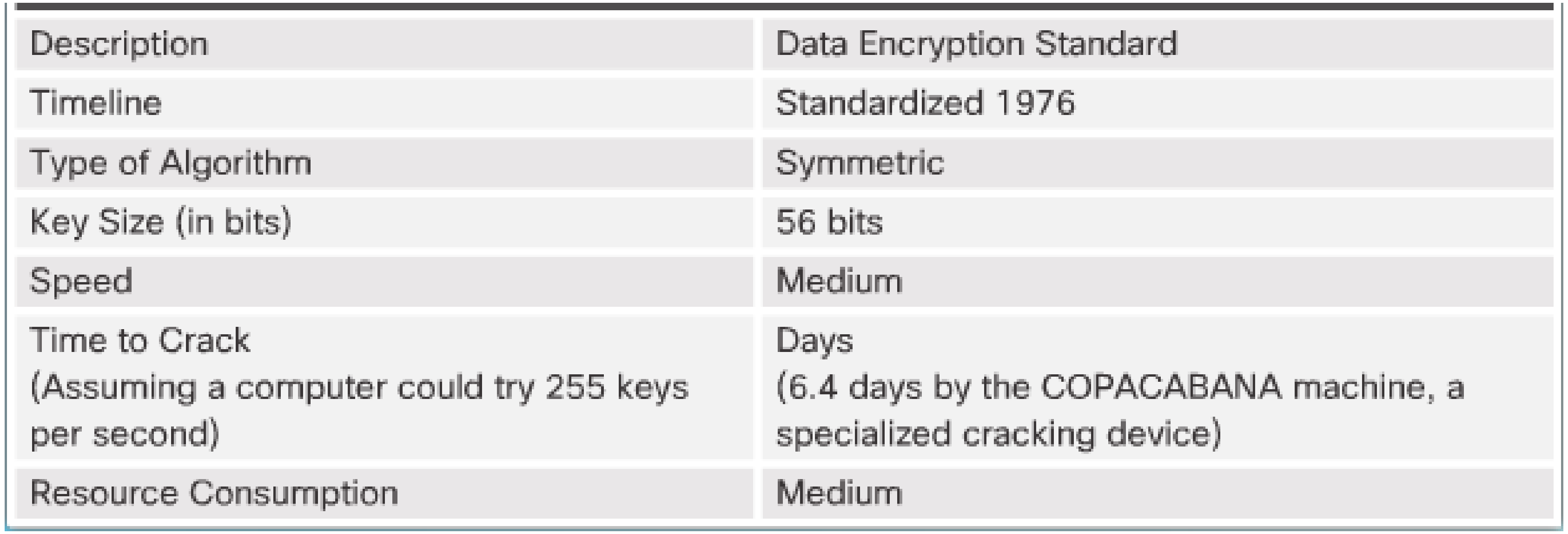
- Administrator decision
- Criteria
  - The algorithm is trusted by the cryptographic community
    - Older ad mature algorithm are more trusted
  - The algorithm adequately protects against brute-force attacks
  - The algorithm supports variable and long key lengths and scalability
  - The algorithm does not have export or import restrictions
    - For example outside of U.S.



	DES	3DES	AES
The algorithm is trusted by the cryptographic community.	Replaced by 3DES	Yes	Ongoing Evaluation
The algorithm adequately protects against brute-force attacks.	No	Yes	Yes

DES Symmetric Encryption

- Well-known, legacy (1976), symmetric, block cipher (64bit), fixed key size of 64bit
  - Too insecure for most of current applications
    - Feasibility of brute-force attacks (or faster types as differential cryptanalysis (DC), linear cryptanalysis, and Davies' attack)
  - Stronger encryption: 56 bits is used for encryption, 8 bits for odd parity of key integrity
  - Weaker encryption: 40bit key, 16 known bit
- Essentially a sequence (16 rounds) of permutations and substitutions of data bits combined with an encryption key



Description	Data Encryption Standard
Timeline	Standardized 1976
Type of Algorithm	Symmetric
Key Size (in bits)	56 bits
Speed	Medium
Time to Crack (Assuming a computer could try 255 keys per second)	Days (6.4 days by the COPACABANA machine, a specialized cracking device)
Resource Consumption	Medium

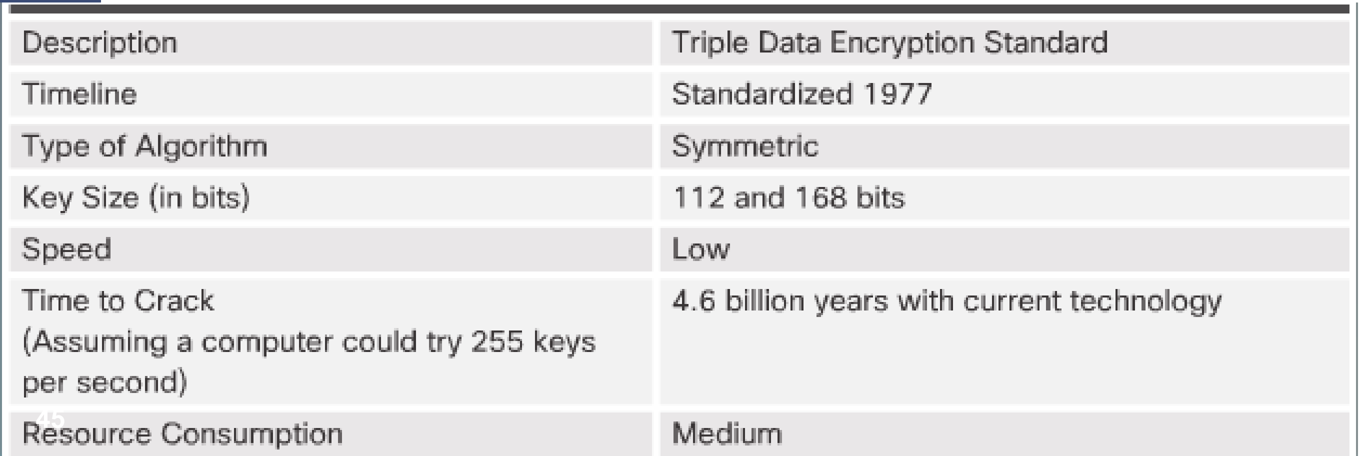
- Summary
  - Insecure, not recommended
  - But if used
    - Change keys frequently
    - Test for a weak key
      - It has 4 weak keys and 12 semi-weak keys
    - Use secure channel for key exchange
    - Use DES in CBC mode of operation

Symmetric Encryption Algorithm

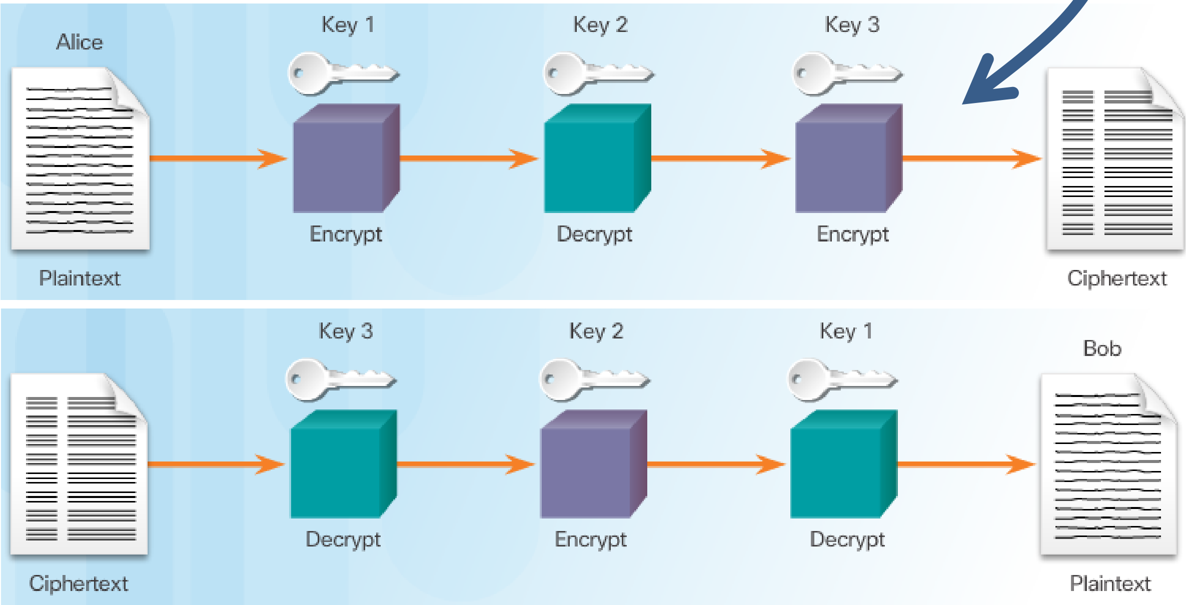
Improved DES => 3DES

- 3DES – improvement of DES algorithm (1977)
  - Called 3DES-Encrypt-Decrypt-Encrypt, Triple DES (TDES), officially the Triple Data Encryption Algorithm (TDEA or Triple DEA),
  - It applies DES three times in a row to a plaintext block
  - Curricula consider it very trustworthy, but resource intensive
    - brute-force attacks are considered infeasible (35years of testing DES and 3DES)
  - => But NIST deprecates 3DES in 2017

- Key size is 64 (56 bits), but
- Uses several keys K1, K2, K3 and three options
  - **Keying option 1,** Key size 168 (3*56bits)
    - All three keys are independent (3DEA), best security
  - **Keying option 2,** Key size 112 (2*56bits)
    - K_1 and K_2 are independent, and K_3 = K_1 (TDEA)
  - **Keying option 3,** Key size 56 (1*56bits)
    - All three keys are identical, worst security



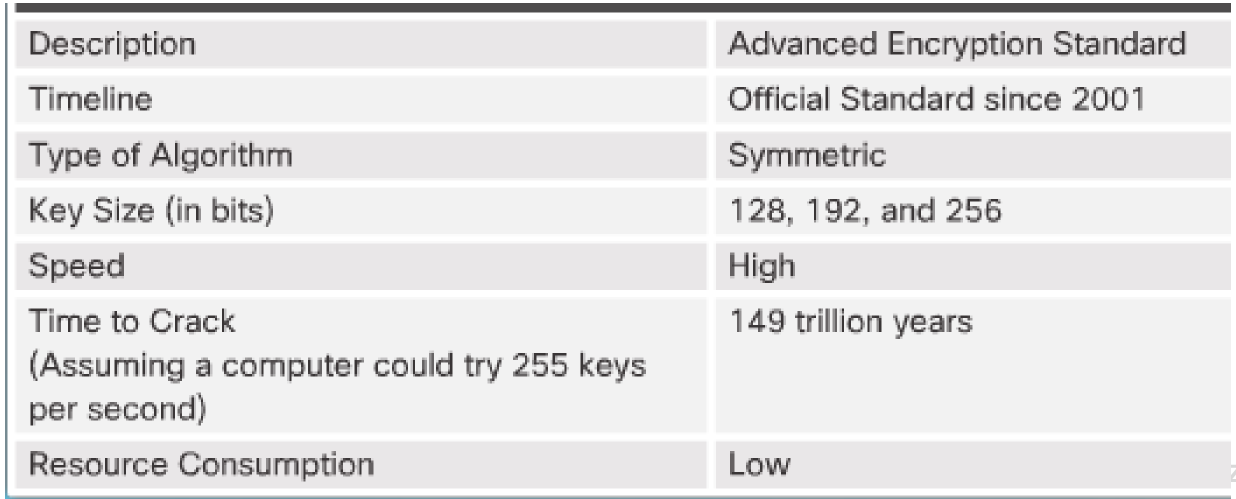
Description	Triple Data Encryption Standard
Timeline	Standardized 1977
Type of Algorithm	Symmetric
Key Size (in bits)	112 and 168 bits
Speed	Low
Time to Crack (Assuming a computer could try 255 keys per second)	4.6 billion years with current technology
Resource Consumption	Medium



Symmetric Encryption Algorithm

Advanced Encryption Standard (AES)

- Proposed as a replacement for DES in 1998
  - Based on Rijndael iterated block cipher
- Uses variable block length and key length (both of 128, 192 or 256-bits)
  - But can be extended in multiples of 32 bits
- As secure (or better) as 3DES, and much faster
  - Support longer keys
  - Supports efficient implementation in hardware or software on a range of processors
    - Suitable for high-throughput, low-latency environments (especially sw. based)
  - But is **younger as DES** ! (less trustworthy)
- Used worldwide
  - Some attack have been published
    - But are not yet computationally feasible

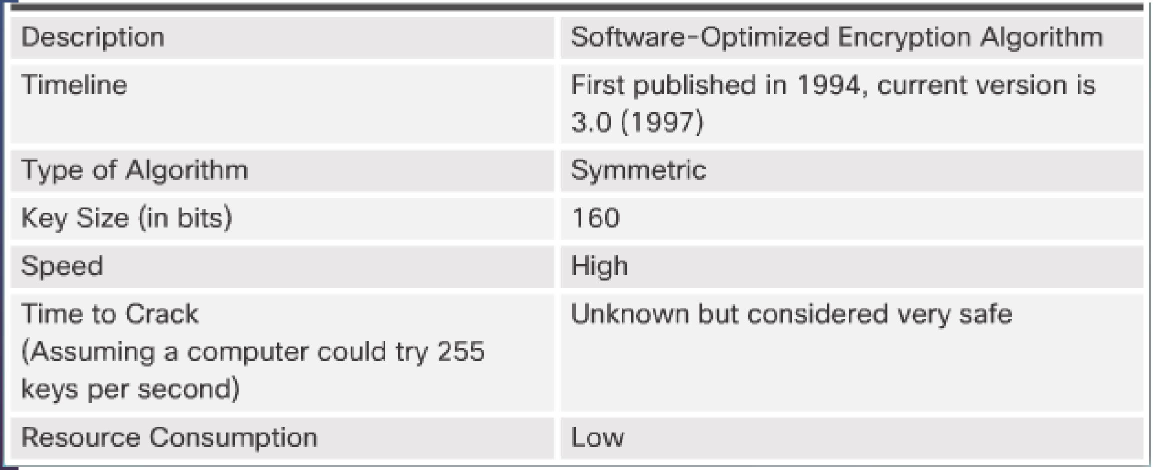


Description	Advanced Encryption Standard
Timeline	Official Standard since 2001
Type of Algorithm	Symmetric
Key Size (in bits)	128, 192, and 256
Speed	High
Time to Crack (Assuming a computer could try 255 keys per second)	149 trillion years
Resource Consumption	Low

Alternate Encryption Algorithms

Software-Optimized Encryption Algorithm (SEAL)

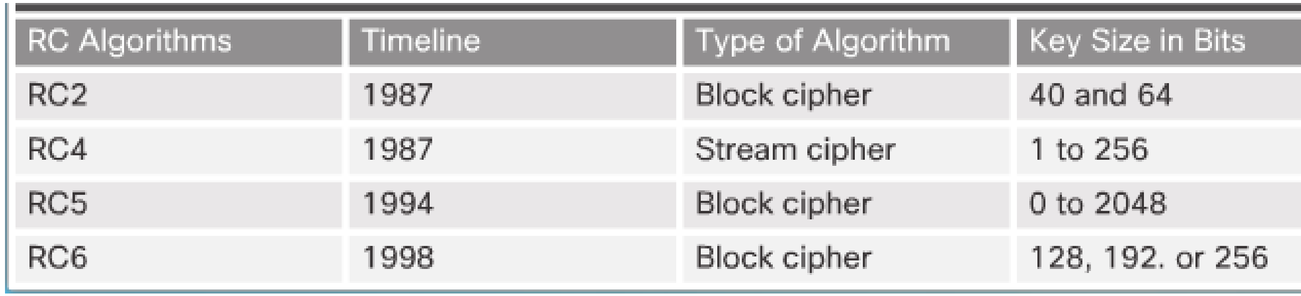
- Alternative to DES/3DES/AES
- Stream cipher with a 160 bit key
- Fast with low CPU overhead
  - But slower initialization phase
- SEAL restrictions for Cisco routers
  - Router and the peer must support IPsec
  - Router and the other peer must run an IOS image that supports encryption
  - Router and the peer must not have hardware IPsec encryption



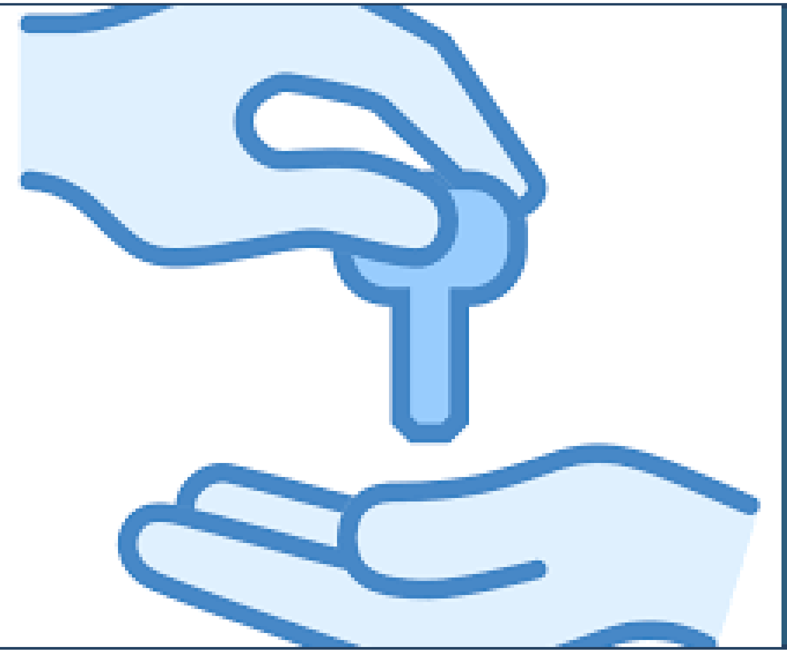
Description	Software-Optimized Encryption Algorithm
Timeline	First published in 1994, current version is 3.0 (1997)
Type of Algorithm	Symmetric
Key Size (in bits)	160
Speed	High
Time to Crack (Assuming a computer could try 255 keys per second)	Unknown but considered very safe
Resource Consumption	Low

RC Algorithms

- Provides very good speed and variable key-length capabilities
- Several RC4 algorithms
  - RC2:
    - replacement for DES
  - RC4:
    - Variable length key, widely used (SSL)
    - Considered secure, but has some bad hackable implementations (WEP)
  - RC6
    - AES opponent



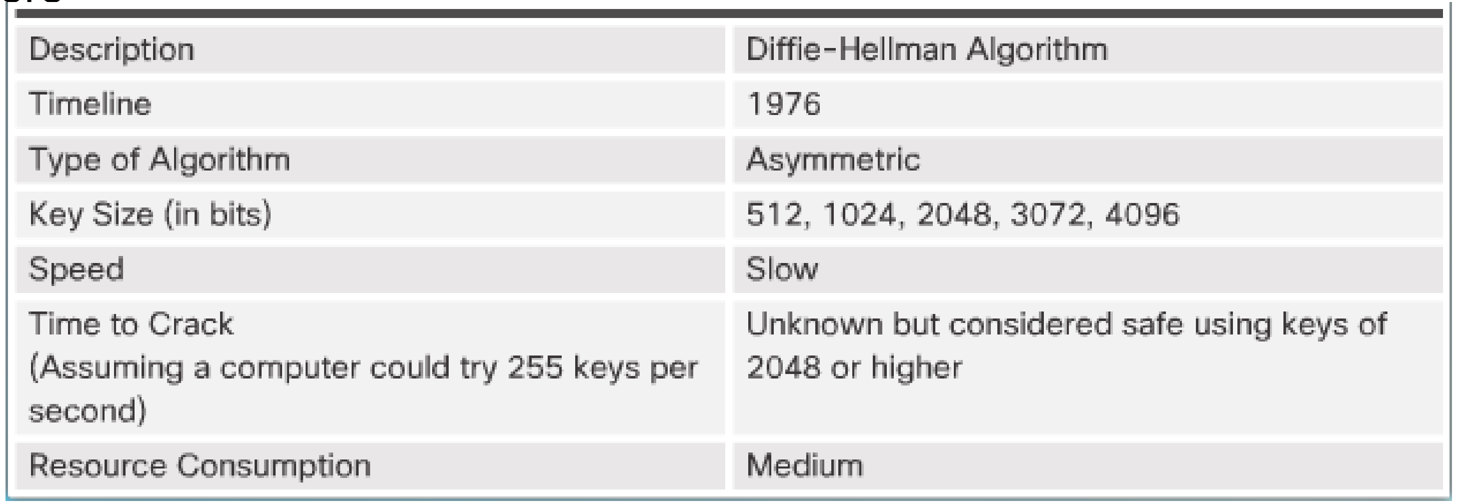
RC Algorithms	Timeline	Type of Algorithm	Key Size in Bits
RC2	1987	Block cipher	40 and 64
RC4	1987	Stream cipher	1 to 256
RC5	1994	Block cipher	0 to 2048
RC6	1998	Block cipher	128, 192. or 256



Diffie-Hellman Key Exchange (for symmetric ciphers)

Diffie-Hellman (DH) Algorithm

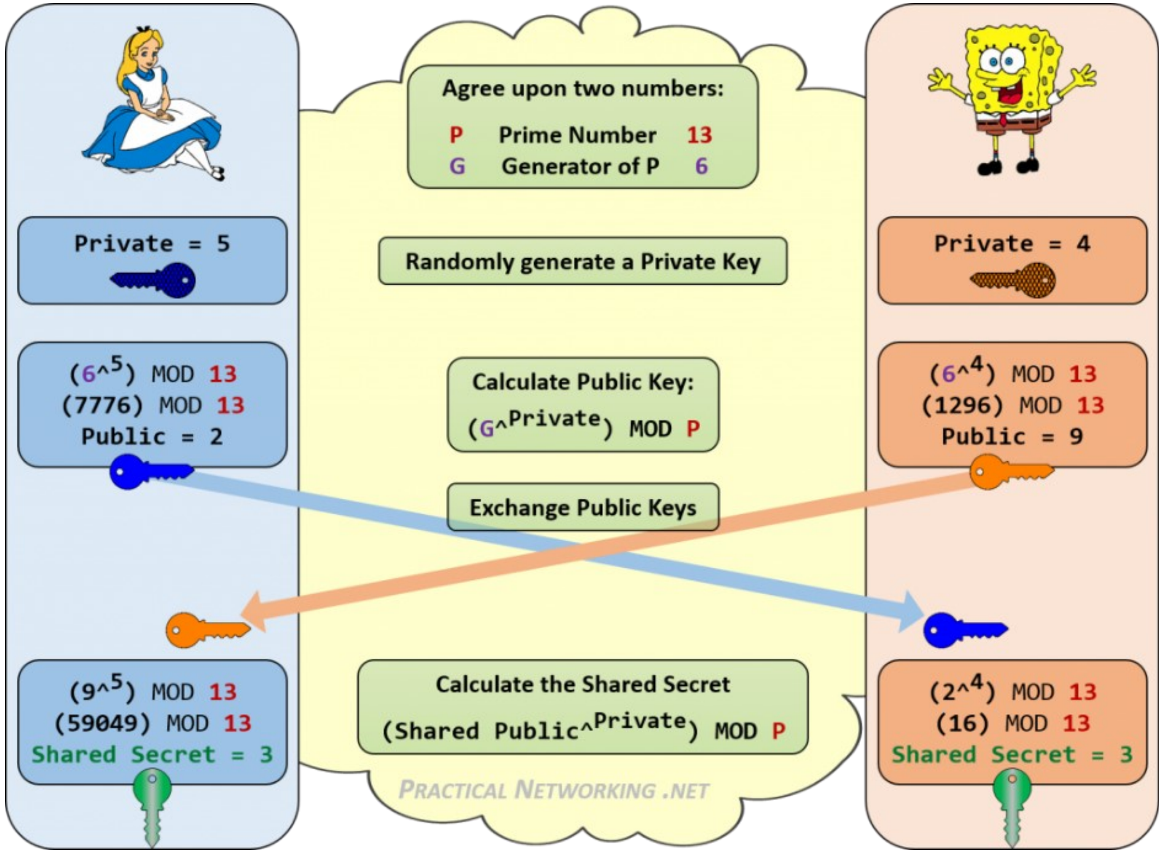
- Remember? The problem (challenge) of symmetric encryption algorithms …
  - … is “*How to exchange the key over an untrusted environment?*”
- DH (Diffie-Hellman (DH) Algorithm)
  - Invented Whitfield Diffie and Martin Hellman in 1976
  - It is not a encryption mechanism
  - Provides automatic and secure key “exchange” method
    - But the shared key is never exchanged over the net
    - Mechanism allows two computers **to generate** an identical shared secret on both systems (using math of large numbers)
      - Without having communicated before
    - Is asymmetric, and slow
  - Commonly used for
    - IPSec VPN, SSL, TLS
      - Initial key handshake using slow and asymmetric DH
      - Encryption then uses fast symmetric algorithm



Description	Diffie-Hellman Algorithm
Timeline	1976
Type of Algorithm	Asymmetric
Key Size (in bits)	512, 1024, 2048, 3072, 4096
Speed	Slow
Time to Crack (Assuming a computer could try 255 keys per second)	Unknown but considered safe using keys of 2048 or higher
Resource Consumption	Medium

Diffie-Hellman Key Exchange

- Several steps
  - 1) Both sites must agree on two shared and non-secret numbers
    - P as a prime number (modulus), typically large
    - G as a base number (generator), typically small
  - 2) Both sites generates one own private number PRIVATE
  - 3) Both sites using G, P and the private number generates a Public number (key)
    - $(G^{PRIVATE})$ MOD P = SHARED PUBLIC KEY
  - 4) Both sites will exchange theirs public keys unencrypted over the net
  - 5) Each site using G, P and an opposite public key to generate shared secret
    - $(SHARED\_PUBLIC^{PRIVATE})$ MOD P = SECRET KEY



: http://www.practicalnetworking.net/series/cryptography/diffie-hellman/

Public Key Cryptography

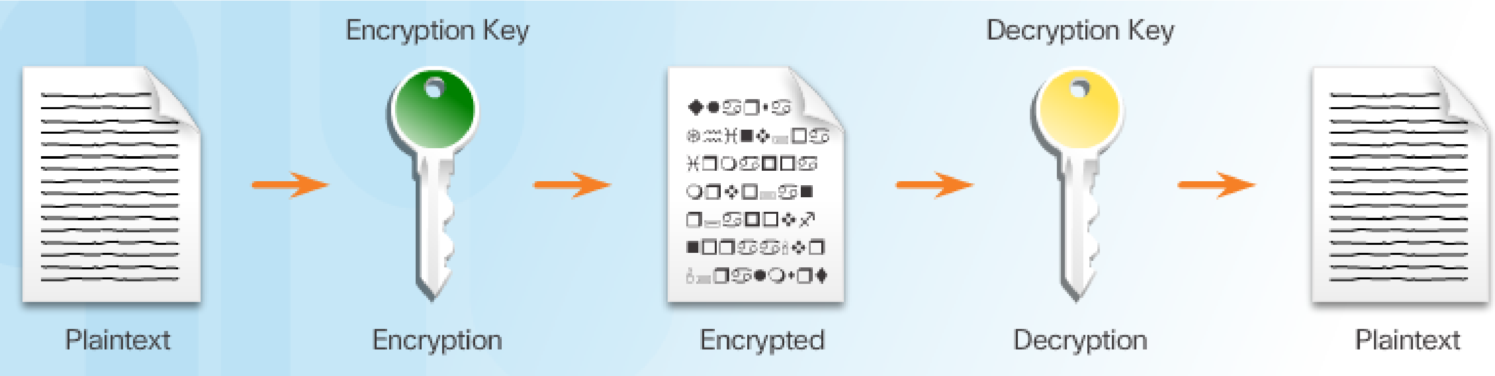
Upon completion of the section, you should be able to:

- Explain the differences between symmetric and asymmetric encryptions and their intended applications.
- Explain the functionality of digital signatures.
- Explain the principles of a public key infrastructure (PKI).

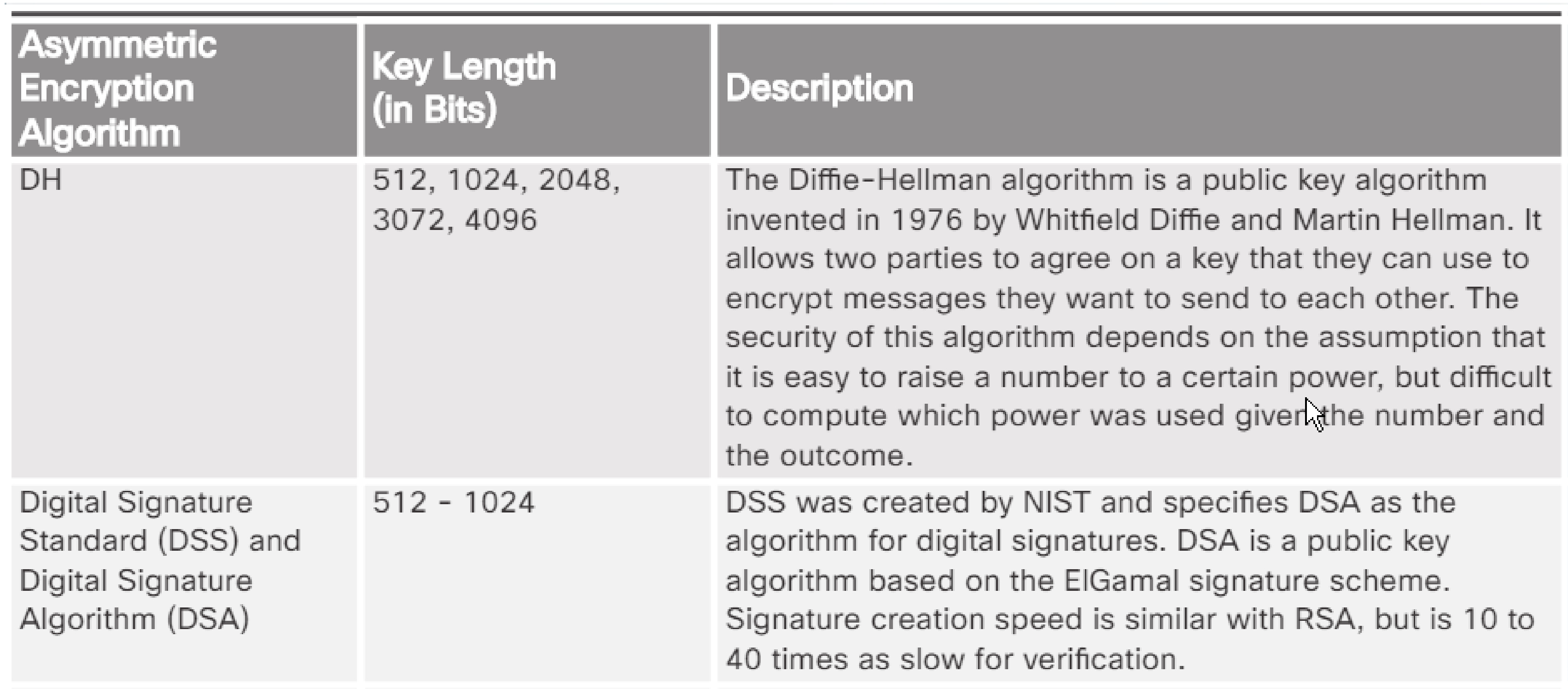
Asymmetric Key Algorithms

- Called public-key algorithms
- Uses separated but matching keys
  - Public key (publicly available)
    - For encryption
  - Private key (kept secure)
    - For decryption
  - Works also in opposite direction
  - There is no option from one key to calculate the other one
  - Typical key length is 512 to 4,096 bits
    - <1024: considered unreliable
    - >1024: considered trusted
- Provides CIA features

- Four well-known protocols that use asymmetric key algorithms:
  - Internet Key Exchange (IKE)
    - for IPSec
  - Secure Socket Layer (SSL) / Transport Layer Security (TLS)
  - Secure Shell (SSH)
  - Pretty Good Privacy (PGP)

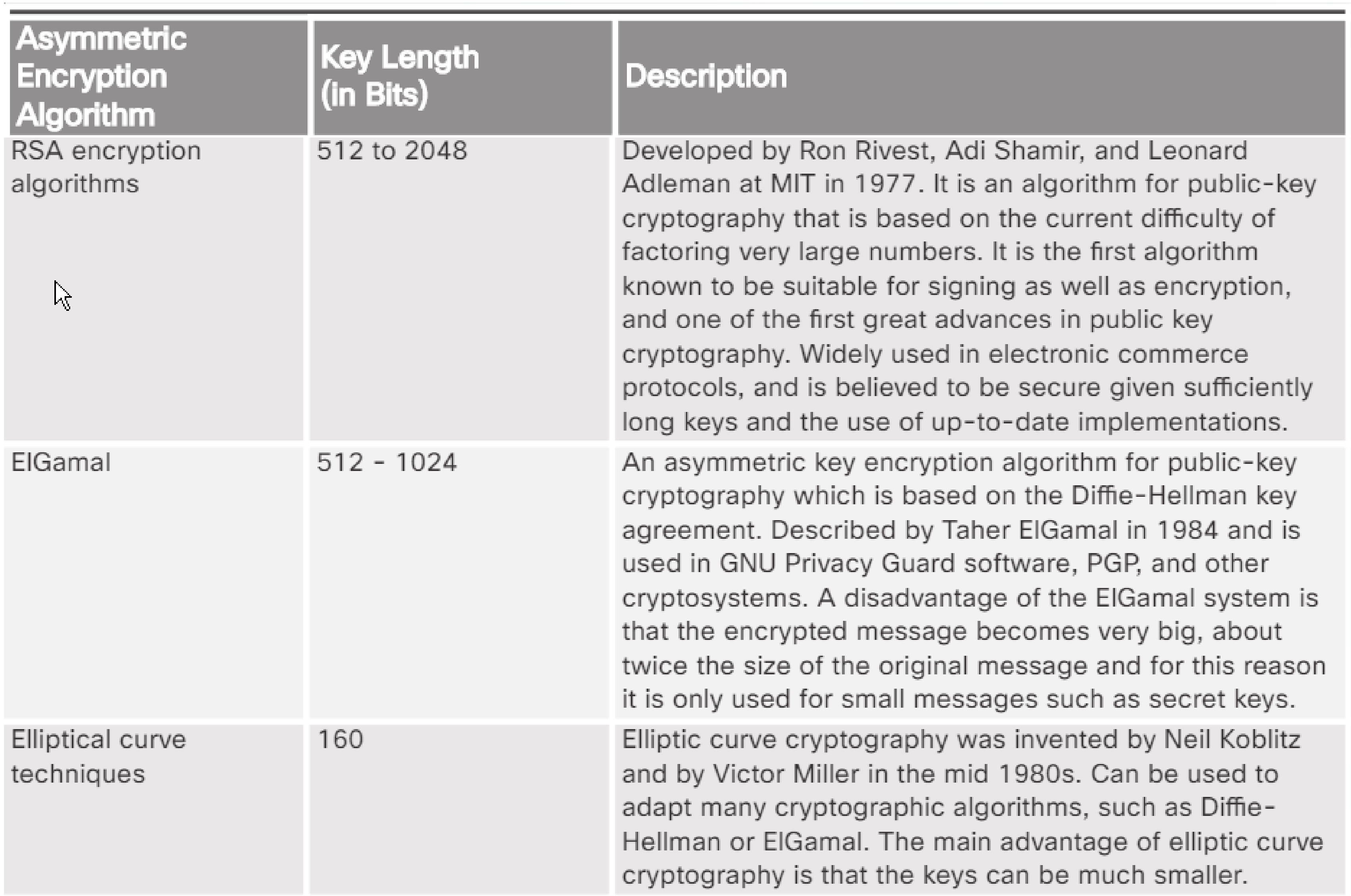


Types of Asymmetric Algorithms (1)



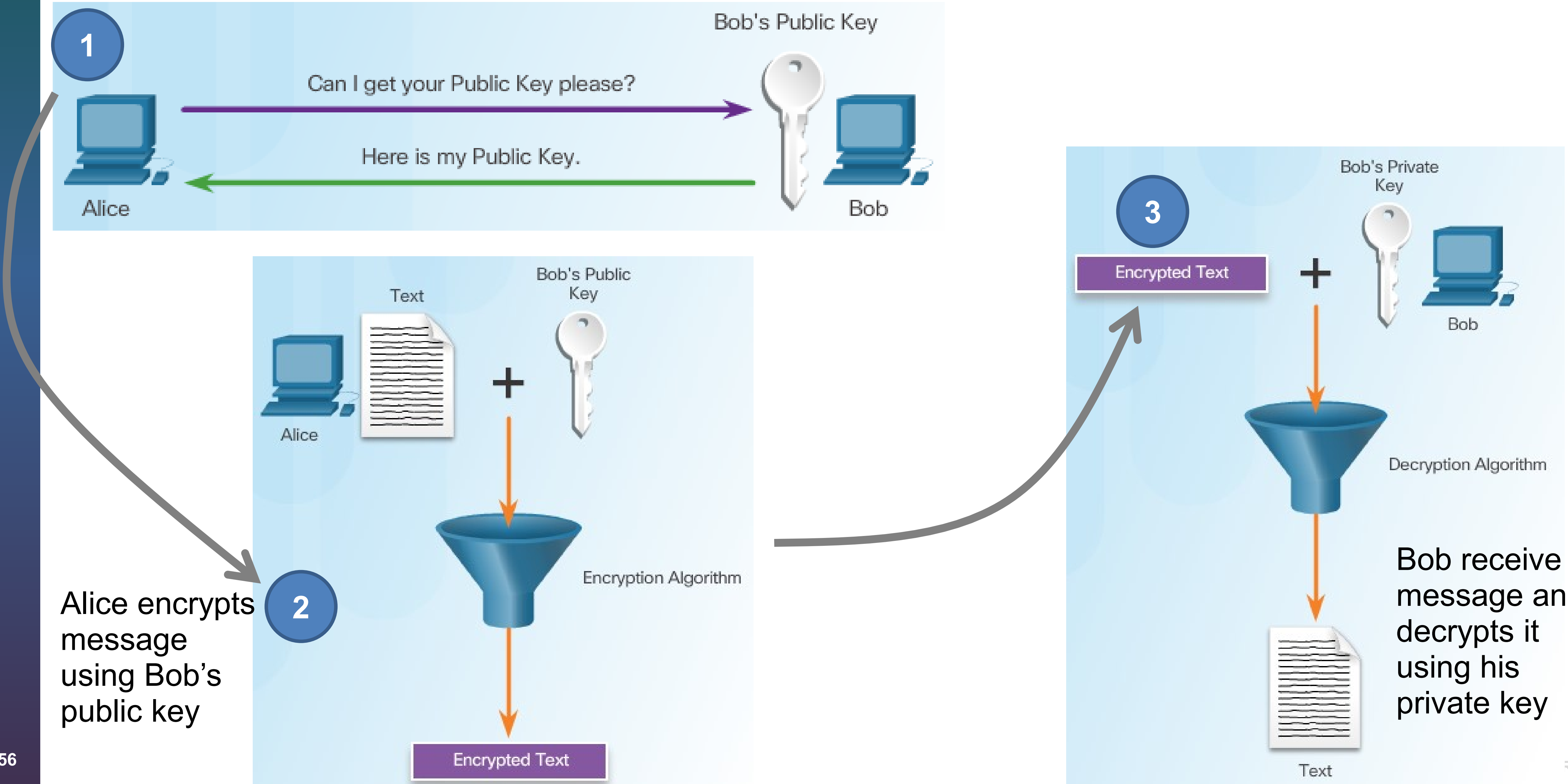
Asymmetric Encryption Algorithm	Key Length (in Bits)	Description
DH	512, 1024, 2048, 3072, 4096	The Diffie-Hellman algorithm is a public key algorithm invented in 1976 by Whitfield Diffie and Martin Hellman. It allows two parties to agree on a key that they can use to encrypt messages they want to send to each other. The security of this algorithm depends on the assumption that it is easy to raise a number to a certain power, but difficult to compute which power was used given the number and the outcome.
Digital Signature Standard (DSS) and Digital Signature Algorithm (DSA)	512 - 1024	DSS was created by NIST and specifies DSA as the algorithm for digital signatures. DSA is a public key algorithm based on the ElGamal signature scheme. Signature creation speed is similar with RSA, but is 10 to 40 times as slow for verification.

Types of Asymmetric Algorithms (2)



Asymmetric Encryption Algorithm	Key Length (in Bits)	Description
RSA encryption algorithms	512 to 2048	Developed by Ron Rivest, Adi Shamir, and Leonard Adleman at MIT in 1977. It is an algorithm for public-key cryptography that is based on the current difficulty of factoring very large numbers. It is the first algorithm known to be suitable for signing as well as encryption, and one of the first great advances in public key cryptography. Widely used in electronic commerce protocols, and is believed to be secure given sufficiently long keys and the use of up-to-date implementations.
ElGamal	512 - 1024	An asymmetric key encryption algorithm for public-key cryptography which is based on the Diffie-Hellman key agreement. Described by Taher ElGamal in 1984 and is used in GNU Privacy Guard software, PGP, and other cryptosystems. A disadvantage of the ElGamal system is that the encrypted message becomes very big, about twice the size of the original message and for this reason it is only used for small messages such as secret keys.
Elliptical curve techniques	160	Elliptic curve cryptography was invented by Neil Koblitz and by Victor Miller in the mid 1980s. Can be used to adapt many cryptographic algorithms, such as Diffie-Hellman or ElGamal. The main advantage of elliptic curve cryptography is that the keys can be much smaller.

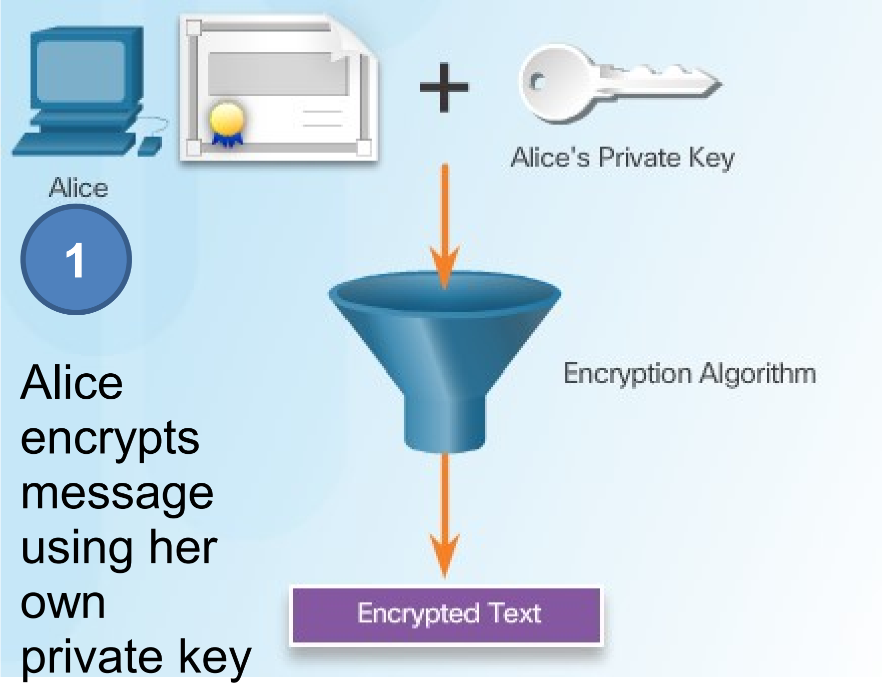
Public Key (Encrypt) + Private Key (Decrypt) = Confidentiality



Alice encrypts message using Bob's public key

Bob receive message and decrypts it using his private key

Private Key (Encrypt) + Public Key (Decrypt) = Authenticity

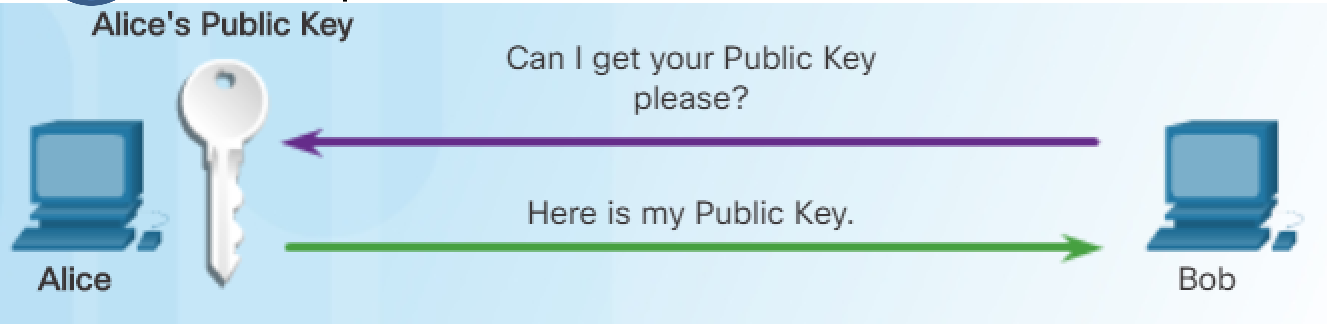


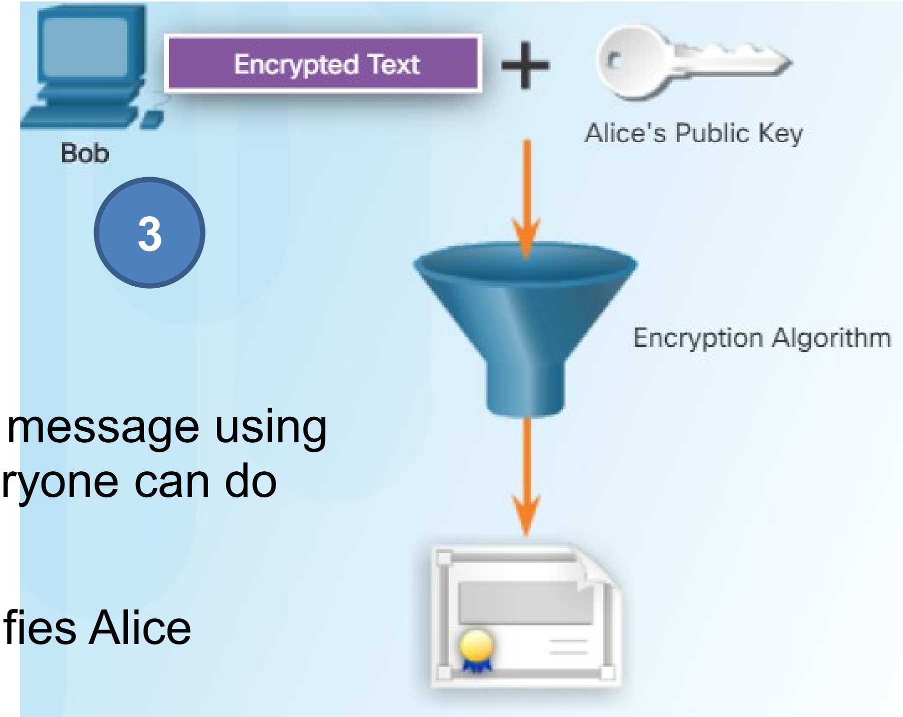
1

Alice encrypts message using her own private key

2

Bob needs Alice's public key to verify Alice. He will request it.





3

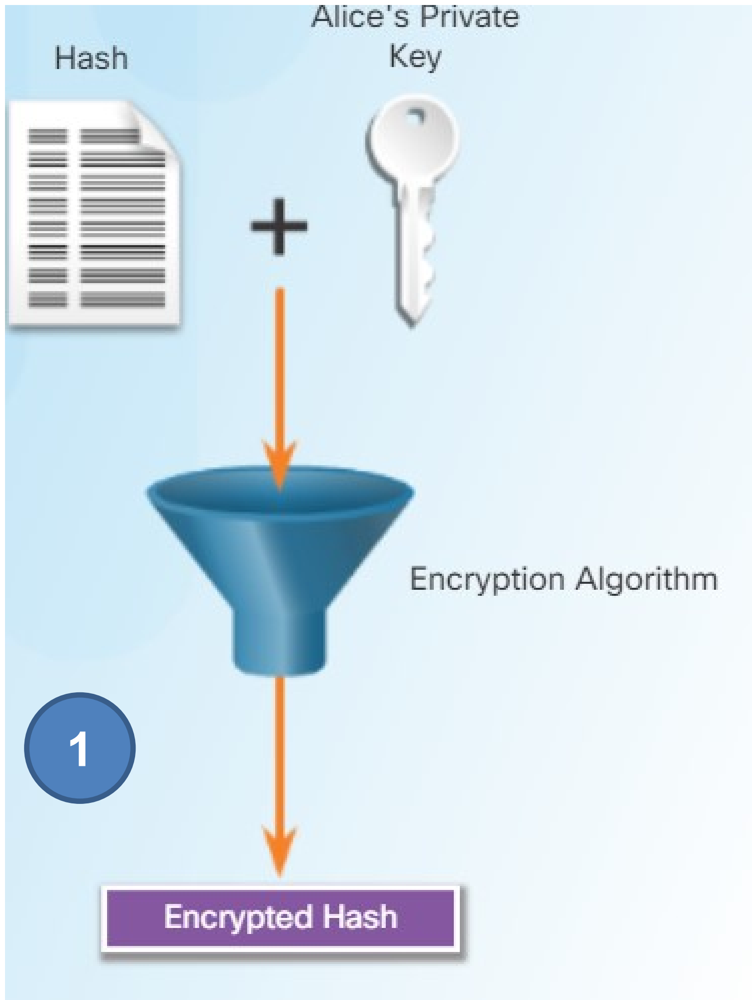
Bob decrypts Alice message using her public key (everyone can do that).

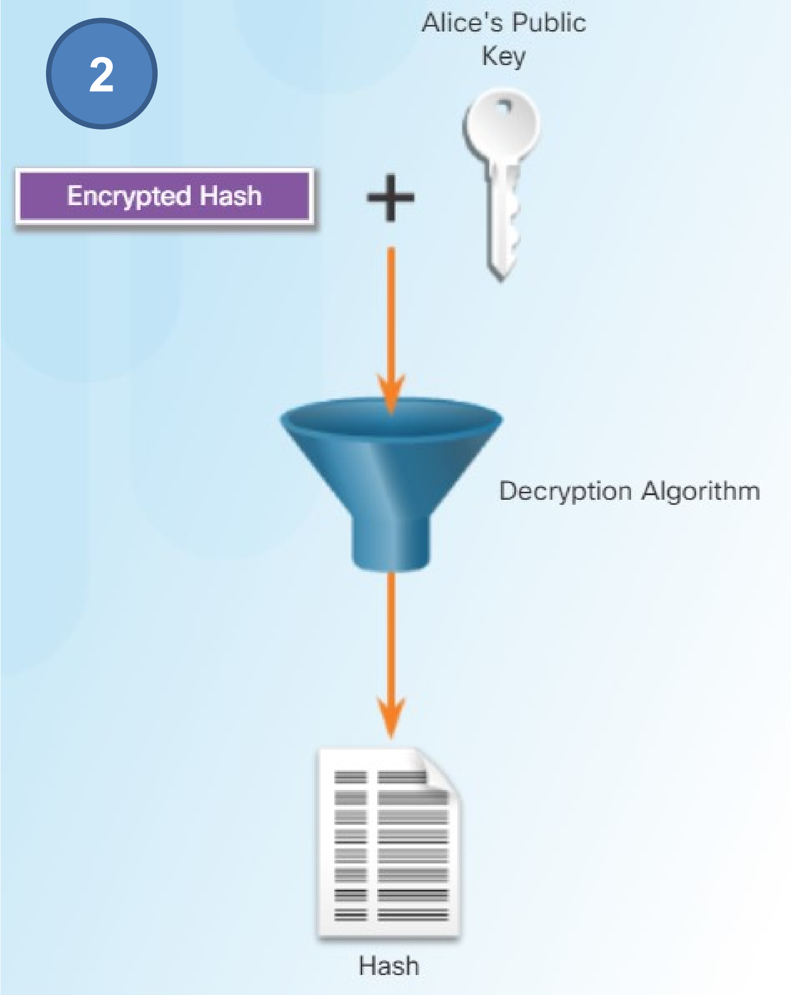
If successful, it verifies Alice

Asymmetric Algorithms - Integrity

- Alice calculates hash across the message
- Encrypt hash using her private key
- Alice attach hash to the message (encrypted or not)

- Bob decrypt Alice hash using her public key
- Bob calculates locally the hash over received message
- If hashes are the same it verifies that message was not modified
- It also verifies Alice

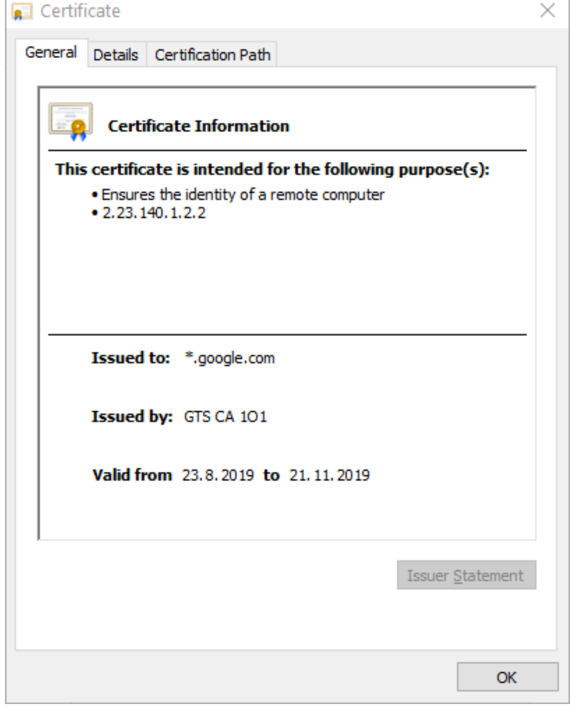




Using Digital Signatures

- Integrity and authenticity features => allows service called ***Digital Signature***
- Digital signature properties:
  - Signature is authentic
    - Provides proof of signer
  - Signature is unalterable
    - Signed document can not be altered
  - Signature is not reusable
    - Is part of a document and can not be moved to another one
  - Signature cannot be repudiated

- Usage
  - Used as a proof of authorship of the content of a document
  - **Digital certificates**
    - Verifies the identity of an organization or individual
    - Authenticates a vendor websites, email senders, …
  - **Code signing**
    - Verifies the integrity of files downloaded from the net
      - Cisco also provides digitally signed IOS images
        - ISR image naming conventions includes “SPA”
      - CLI: **show software authenticity flash:NAME**
    - Authenticates and verifies the identity of the site (using digital certs) of issuer/publisher



Certificate Information

This certificate is intended for the following purpose(s):

- Ensures the identity of a remote computer
- 2.23.140.1.2.2

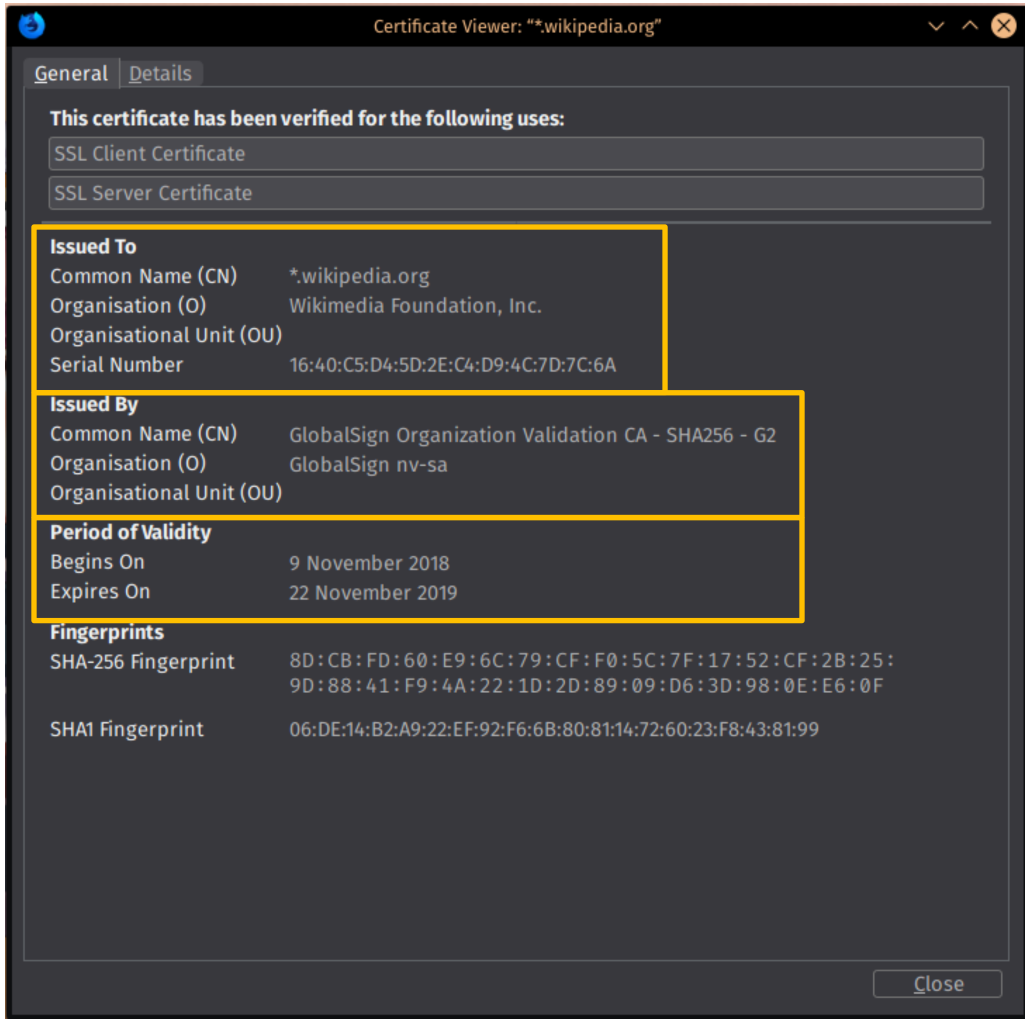
Issued to: *.google.com

Issued by: GTS CA 1O1

Valid from 23.8.2019 **to** 21.11.2019

Digital Certificate (DC)

- An electronic document used to prove the ownership of a public key
- Includes
  - information about the key
  - information about the identity of its owner (called the subject)
    - A person, computer/server
  - digital signature of an entity that has verified the certificate's contents (called the issuer)
  - If is DC valid and we trust issuer’s we may use it
- Many DC types
  - TLS/SSL server certificate
  - TLS/SSL client certificate
  - Email certificate
  - Code signing certificate (to validate signatures on programs)
  - Qualified certificate (for electronic signature of individuals)
  - Root certificate (a self-signed certificate used to sign other certificates)
  - Intermediate certificate (used to sign other certs)
  - End-entity or leaf certificate (cannot be used to sign other certificates)
  - Self-signed certificate (a subject matches its issuer)



This certificate has been verified for the following uses:

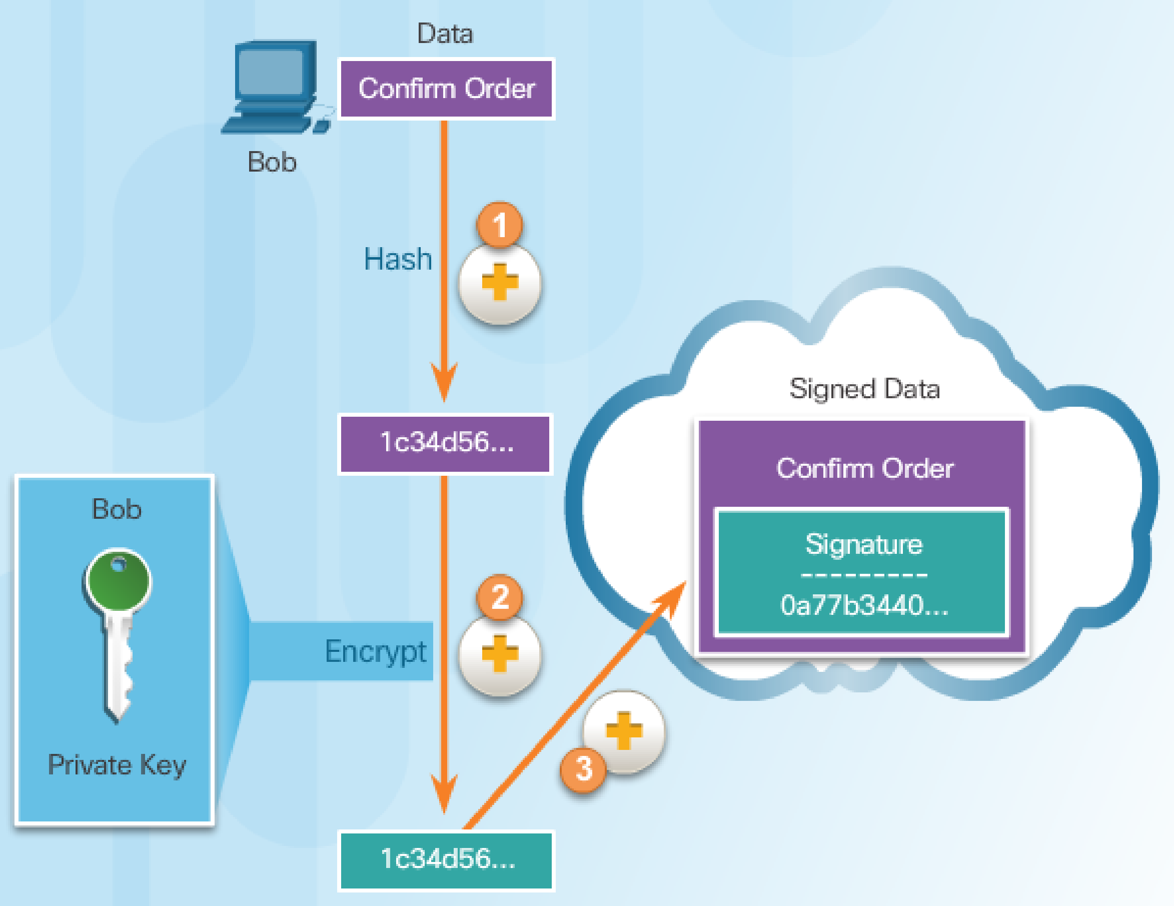
SSL Client Certificate

SSL Server Certificate

Issued To	
Common Name (CN)	*.wikipedia.org
Organisation (O)	Wikimedia Foundation, Inc.
Organisational Unit (OU)	
Serial Number	16:40:C5:D4:5D:2E:C4:D9:4C:7D:7C:6A
Issued By	
Common Name (CN)	GlobalSign Organization Validation CA - SHA256 - G2
Organisation (O)	GlobalSign nv-sa
Organisational Unit (OU)	
Period of Validity	
Begins On	9 November 2018
Expires On	22 November 2019
Fingerprints	
SHA-256 Fingerprint	8D:CB:FD:60:E9:6C:79:CF:F0:5C:7F:17:52:CF:2B:25: 9D:88:41:F9:4A:22:1D:2D:89:09:D6:3D:98:0E:E6:0F
SHA1 Fingerprint	06:DE:14:B2:A9:22:EF:92:F6:6B:80:81:14:72:60:23:F8:43:81:99

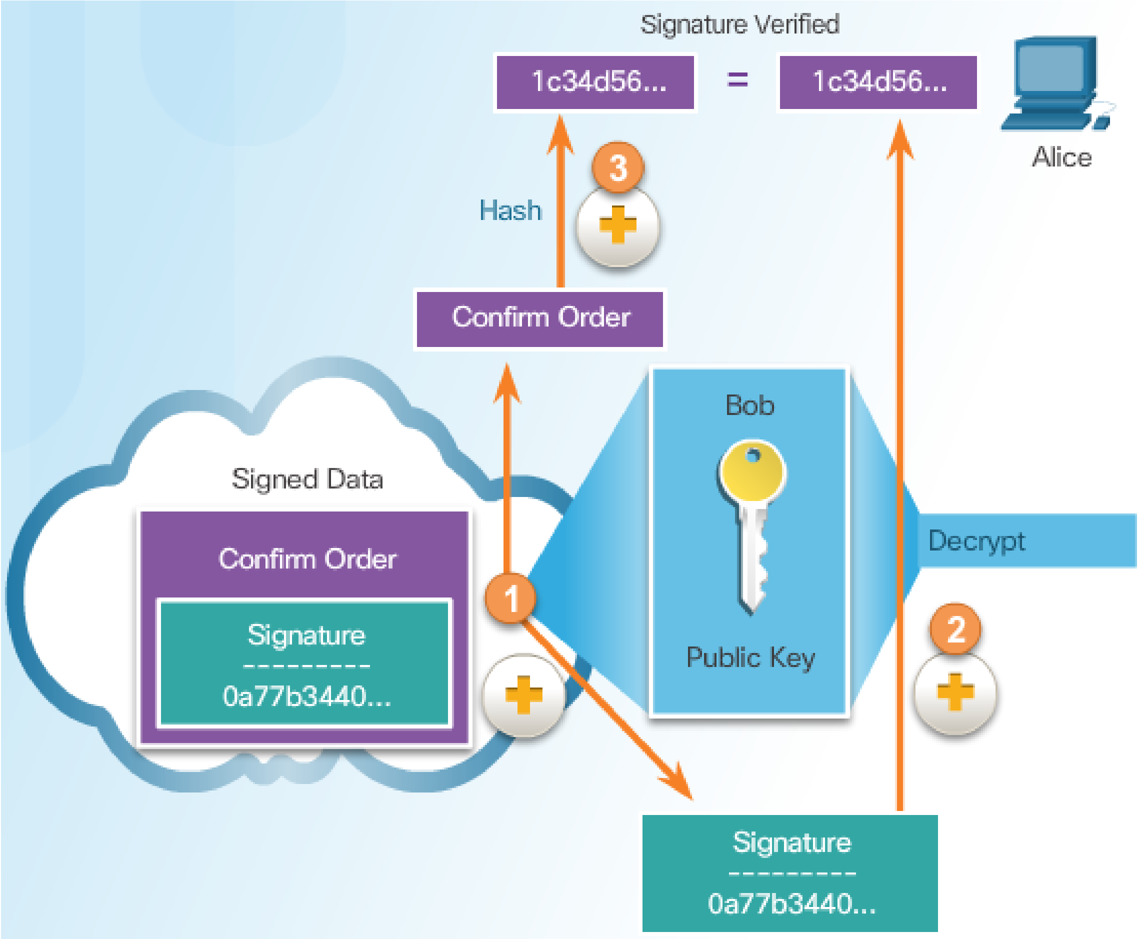
Wiki client/server cert

Using Digital Certificates



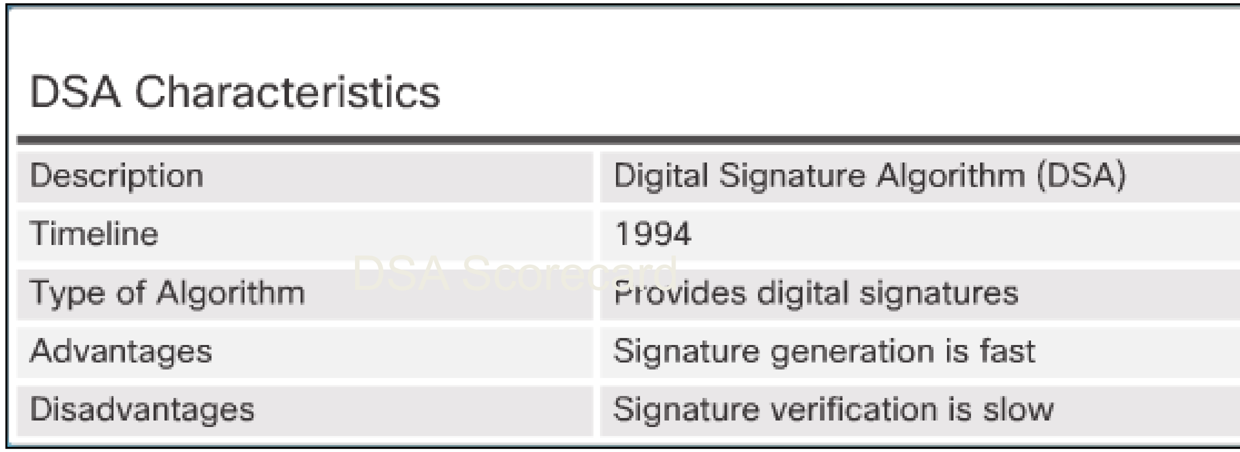
Receiving a Digital Certificate

Sending a Digital Certificate



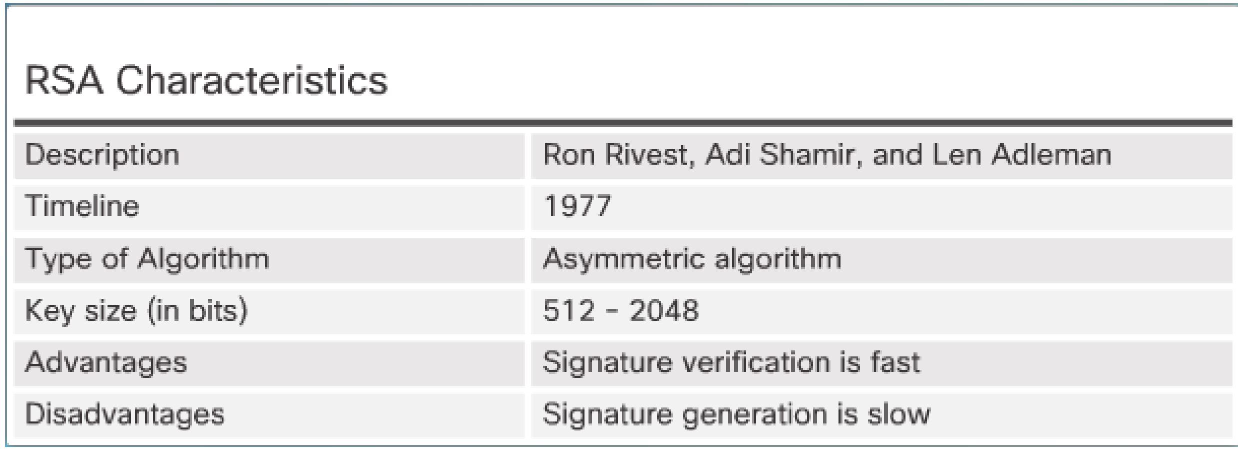
Digital Signature Algorithms

- Three algorithms
  - **Digital Signature Algorithm (DSA)**
    - original standard for
      - generating public and private key pairs
      - generating and verifying digital signatures.
  - **Rivest-Shamir Adelman Algorithm (RSA) digital signature algorithm**
    - asymmetric algorithm for generating and verifying digital signatures.
  - **Elliptic Curve Digital Signature Algorithm (ECDSA)**
    - a newer variant of DSA
    - provides
      - digital signature authentication and non-repudiation
      - computational efficiency
      - small signature sizes
      - minimal bandwidth



DSA Characteristics

Description	Digital Signature Algorithm (DSA)
Timeline	1994
Type of Algorithm	Provides digital signatures
Advantages	Signature generation is fast
Disadvantages	Signature verification is slow



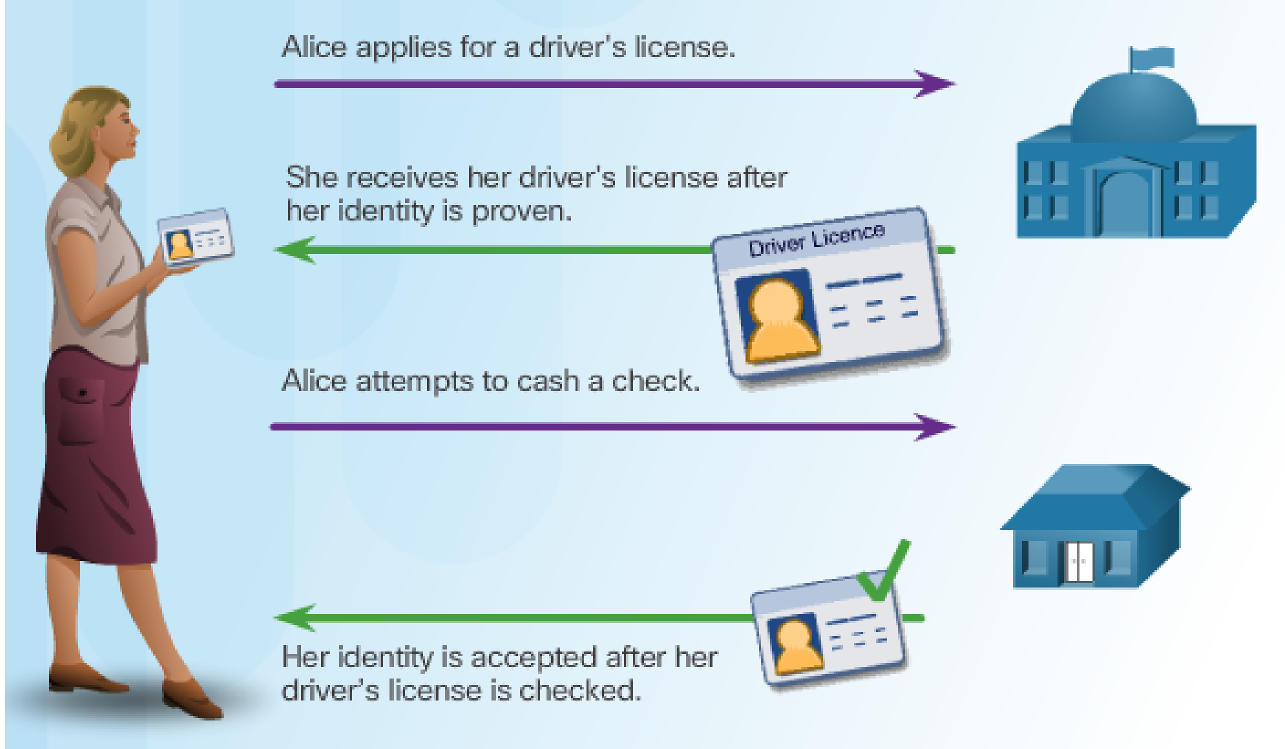
RSA Characteristics

Description	Ron Rivest, Adi Shamir, and Len Adleman
Timeline	1977
Type of Algorithm	Asymmetric algorithm
Key size (in bits)	512 - 2048
Advantages	Signature verification is fast
Disadvantages	Signature generation is slow

Public Key Infrastructure

Public Key Infrastructure

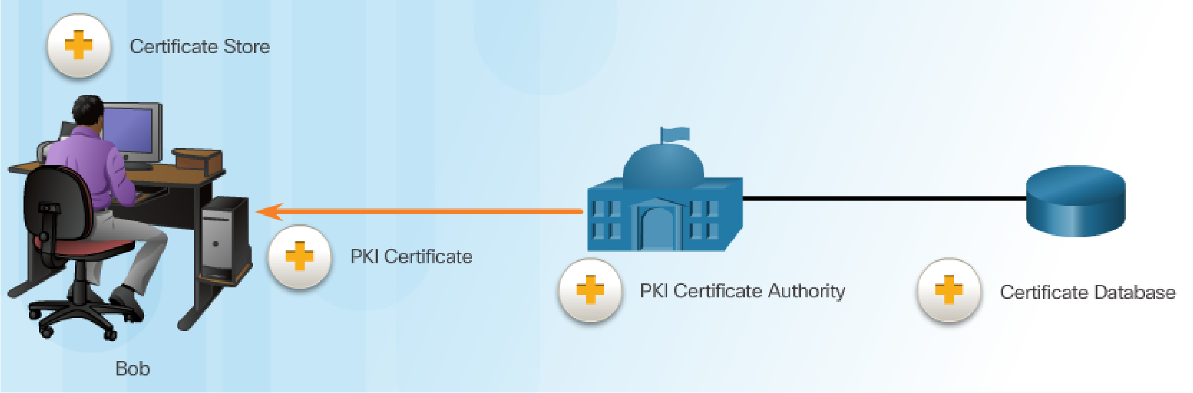
- PKI
  - Solve the problem of secure exchange of identity information
  - Using concept of authority
    - Neutral, commonly trusted third party, a.k.a ***certificate authority***
      - Others accept its credentials
    - It does in-depth investigation of authenticity
    - And it issues digital certificates
- PKI is the framework inspired by legacy authenticity procedures
  - Support large-scale deployment
  - Consists of:
    - hardware, software, people, policies, and procedures
    - needed to create, manage, store, distribute, and revoke digital certificates

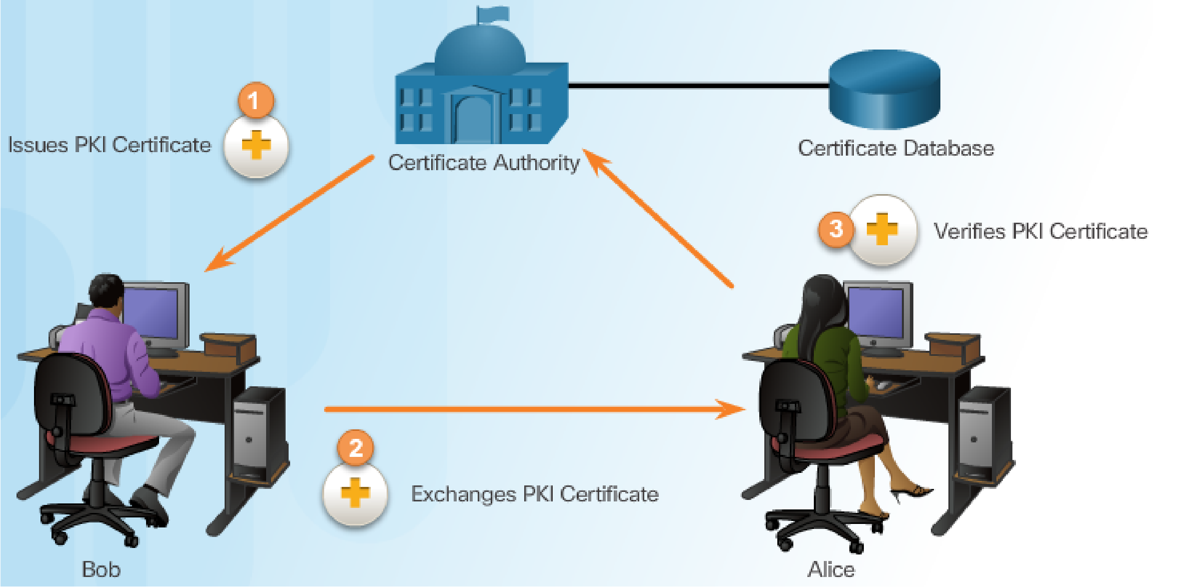

Alice applies for a driver's license.
She receives her driver's license after her identity is proven.
Driver Licence
Alice attempts to cash a check.
Her identity is accepted after her driver's license is checked.

- Without PKI
  - We can achieve confidentiality
  - But not authenticity

PKI Framework components

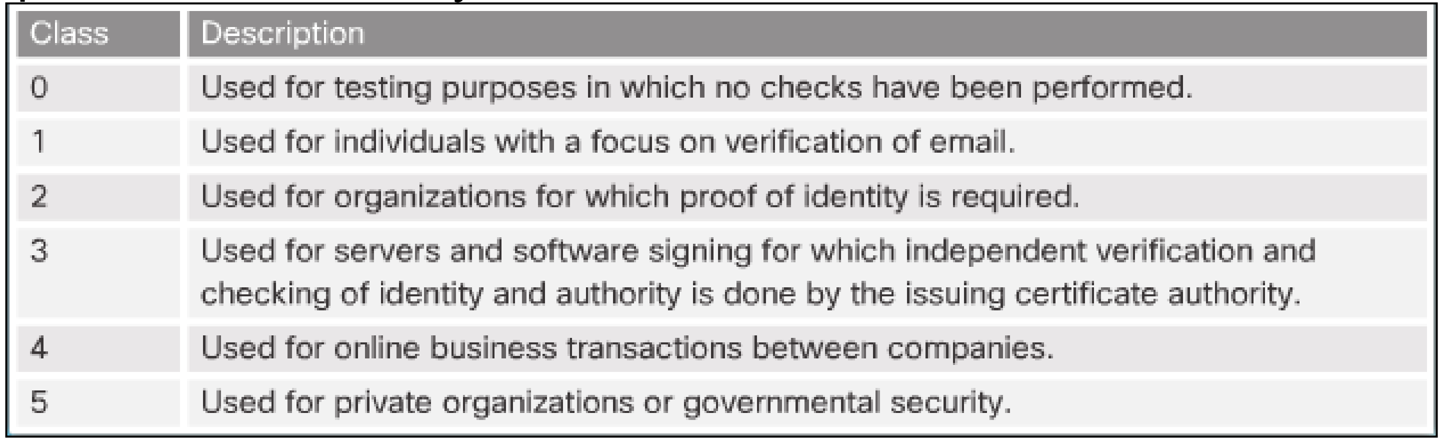
- **Certificate store**
  - Resides on a local computer
  - Store issued certificates and private keys
- **Certificate authority (CA)**
  - Globally trusted third party (company)
  - Locally trusted (enterprise or state)
  - Verifies identity
  - Issues PKI certificates to entities and individuals
  - Digitally signs these certificates
    - using its private key
- **Registration authority (RA)**
  - A subordinate CA
  - certified by a root CA to offload some CA activites
- **Certification database**
  - Stores all certificates approved by CA





Certificate Authorities and CA servers

- CAs may issue certificates of a number of classes
  - Class determines how trusted a certificate is
  - Higher the class number, the more trusted the certificate is
    - Class identifies how rigorous the procedure of identity verification was

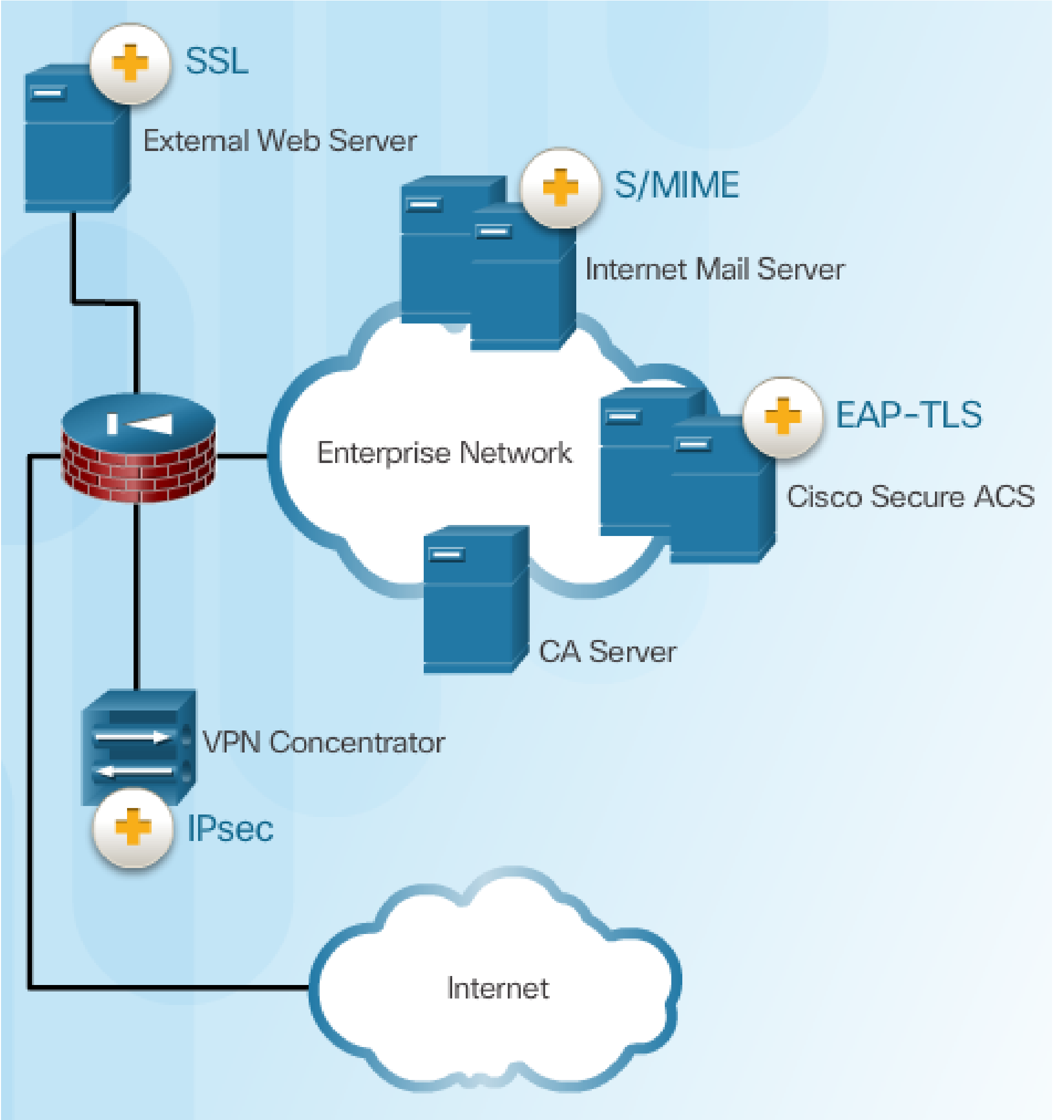


Class	Description
0	Used for testing purposes in which no checks have been performed.
1	Used for individuals with a focus on verification of email.
2	Used for organizations for which proof of identity is required.
3	Used for servers and software signing for which independent verification and checking of identity and authority is done by the issuing certificate authority.
4	Used for online business transactions between companies.
5	Used for private organizations or governmental security.

- CA service may be offered through CA servers.
- CAS offers
  - Management of certificate enrolment and all required processes
    - Generates a root certificate for digitally signing other certificates
    - Allows to create and store asymmetric PKI key pairs, signing or validating anything that depends on a PKI, signing firmware updates, code and etc.
  - Actually exists many vendor solutions with many features (Symantec Group (VeriSign), Comodo, Go Daddy Group, GlobalSign, DigiCert, and others)

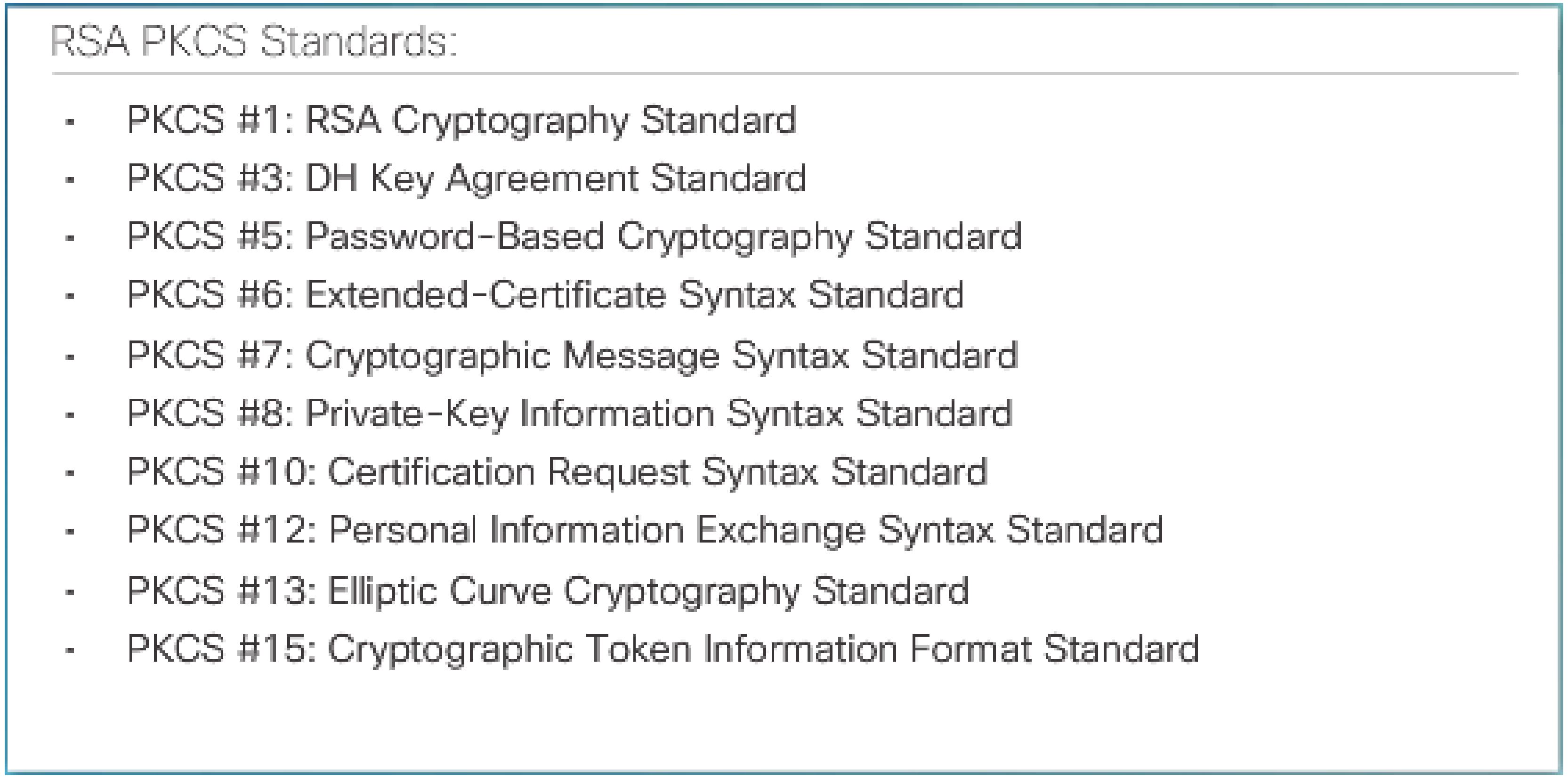
Interoperability of Different PKI Vendors

- PKI infrastructure
  - Can be deployed using different vendors entities
  - Requires interconnection with different supporting services (LDAP, X.500)
  - Issues of interoperability emerged
- IETF reacts with RFC 3467 standard
  - Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework
  - Promotes and standardizes PKI on the Internet
    - Defines basic PKI formats
      - as the certificate and certificate revocation list (CRL) format
  - X.509 certificate format is already extensively used and supported
    - IPSec VPN authentication, router/switches auth, secure web servers supports X509 SSL/TLS, email agents …



Public-Key Cryptography Standards

- PKCS = group of public-key cryptography standards
  - https://en.wikipedia.org/wiki/PKCS

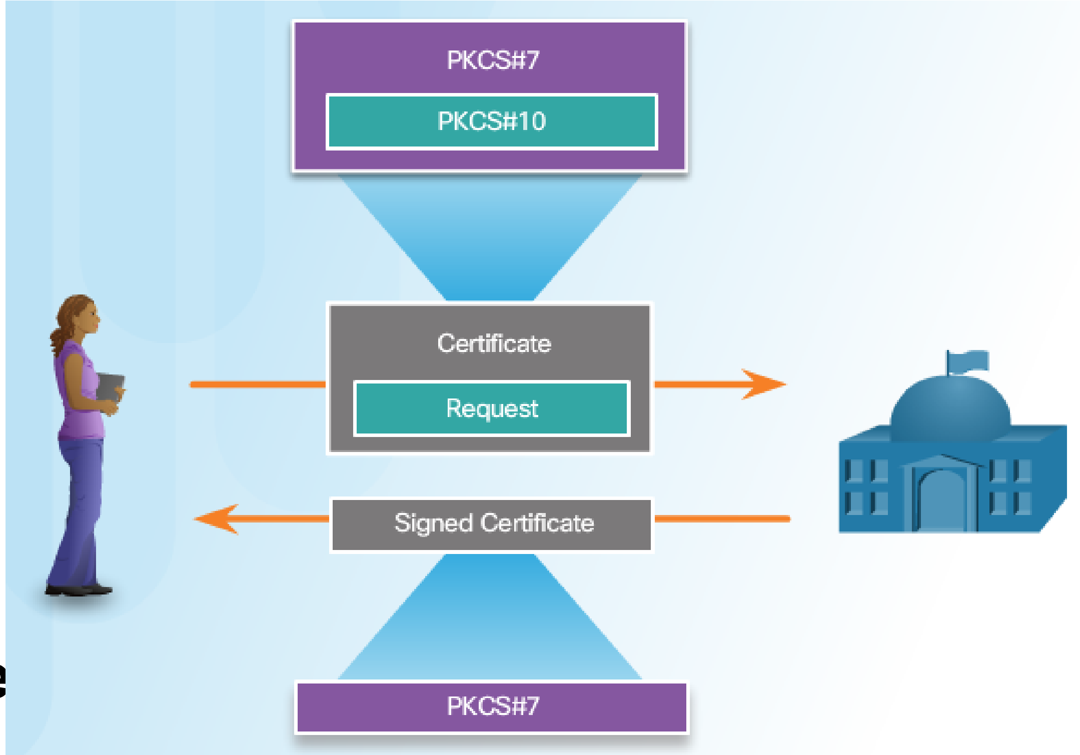


RSA PKCS Standards:

- PKCS #1: RSA Cryptography Standard
- PKCS #3: DH Key Agreement Standard
- PKCS #5: Password-Based Cryptography Standard
- PKCS #6: Extended-Certificate Syntax Standard
- PKCS #7: Cryptographic Message Syntax Standard
- PKCS #8: Private-Key Information Syntax Standard
- PKCS #10: Certification Request Syntax Standard
- PKCS #12: Personal Information Exchange Syntax Standard
- PKCS #13: Elliptic Curve Cryptography Standard
- PKCS #15: Cryptographic Token Information Format Standard

Simple Certificate Enrollment Protocol

- PKI technology
  - Actually is extensively used
  - Became defacto basis for standardized security
  - Arises requirements on scalable certificates management suitable for PKI clients and CA servers
    - Whom perform enrolment, revocation, lifecycle…
  - Previously almost processed manually by admins
    - Not suitable
      - For large deployments
      - For easy and electronic use
- => IETF is preparing the **Simple Certificate Enrollment Protocol** (SCEP) (only draft)
  - Effort to make issuing and revocation of digital certificates as scalable as possible
  - Allows simplified means of handling certificates for large-scale implementation
    - secure issuance of certificates to network devices

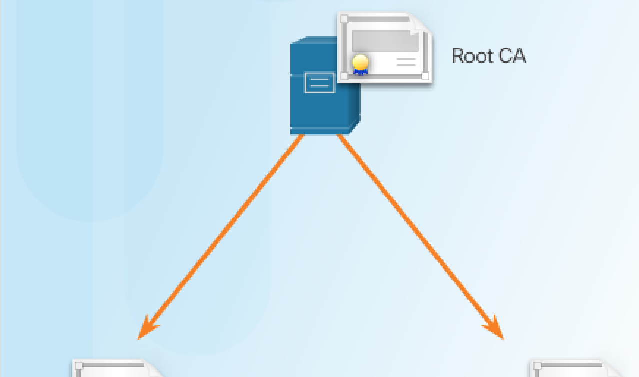


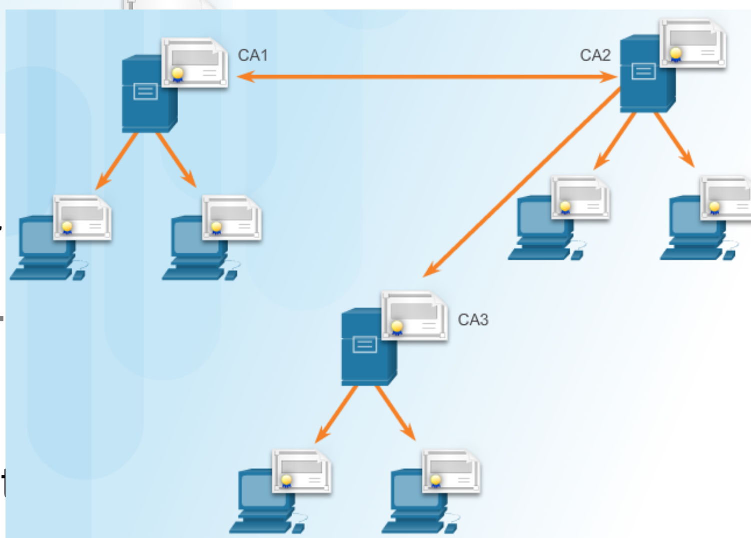
PKI deployment models

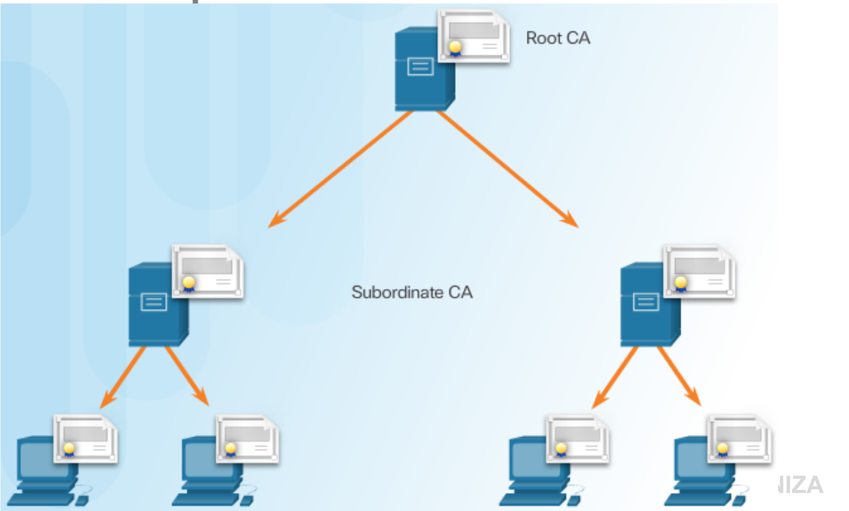
- PKI forms a topologies of trusts
  - **Single-Root PKI Topology**
    - The simplest one: usually one organization with Single CA (Root CA): Issues all certs
    - Benefit - simple
    - Minus: no one trust its certs, not scalable solution, Single point of Failure

HYBRID - combination

  - **Cross-certified CA topologies**
    - Peer-to-peer model
      - => CA establish trust with other CAs: by cross-certifying theirs certificat
      - Trusted CAs trust each other and theirs certs
    - Solution provides redundancy, no Single PoFail
  - **Hierarchical CA topologies**
    - CA organized on levels
      - Root CA: highest CA
        - Issues certs to end users and subordinate CAs
        - Establish community of trust
      - Benefits: increased scalability and manageability
      - Minus: chain of signing CA can be sometimes difficult

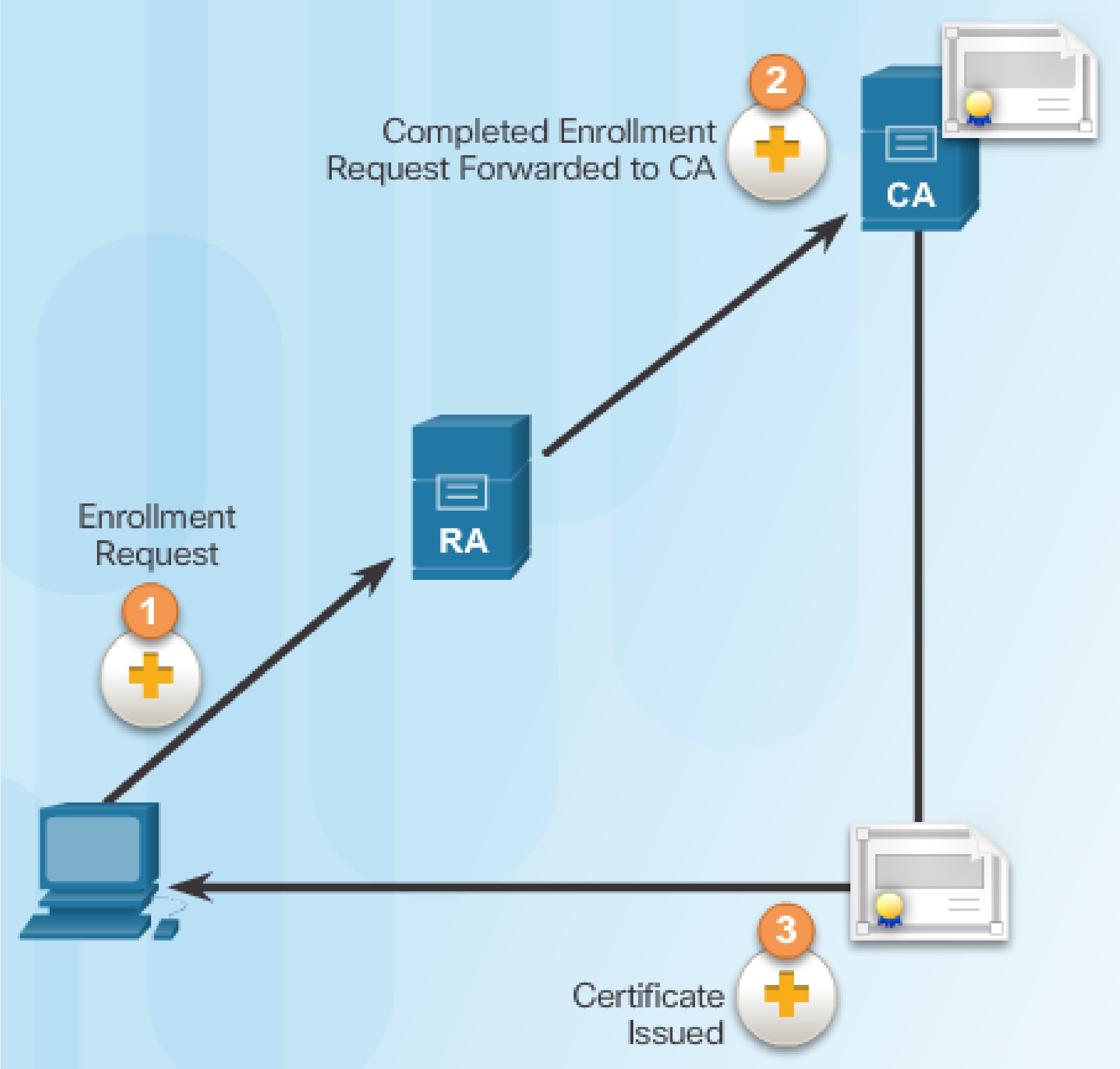






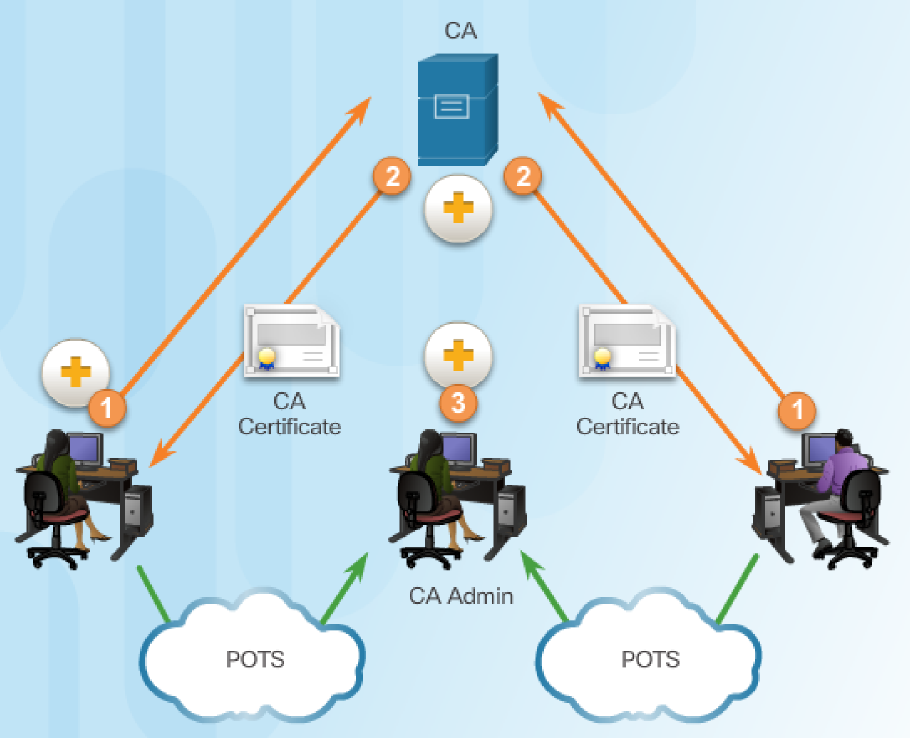
Registration Authority - RA

- Reduce the burden of CAs
  - where is high volume of cert transaction
  - RA can accept requests for enrollment in the PKI
    - Identification and authentication of subscribers
    - Accept registration and certificate requests (1)
    - Forwards requests to CA (2)
    - Does not sign or issues certs,
      - only CA may do that (3)
- May handle three tasks
  - Authentication of users when they enroll with the PKI
  - Key generation for users that cannot generate their own keys
  - Distribution of certificates after enrollment

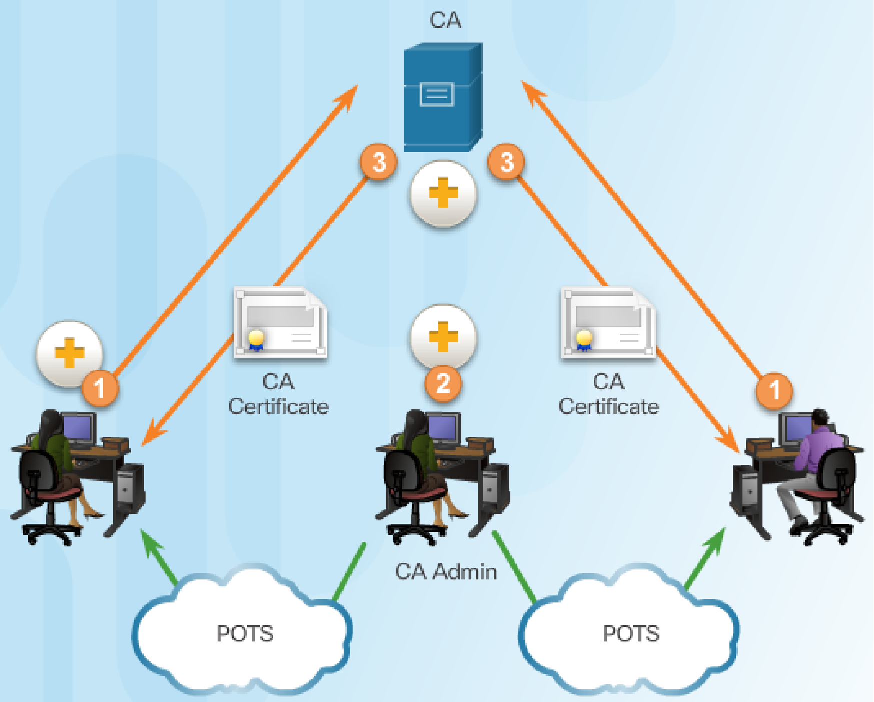


Digital Certificates and CAs - processes

- 1-2) User has to securely obtain CA's public key (self-signed certificate)
  - Key verifies all the certificates issued by the CA
    - it is essential for PKI, only root CA issues it
  - Downloaded in-band, stored locally
- 3) User authenticates CA's certificate out-of-band to CA admin (personally, call …)
  - Serial number



- 1) Users submit his certificate requests
- 2) CA server receive requests
  - CA admin call to users for confirmation
  - Add registration data, digitally sign it, issues certificates
- 3) Users download and install theirs certs
  - manually or using SCEP



Certificate Enrollment, Authentication, and Revocation

- All systems that leverage the PKI must have the CA's public key, which is called the self-signed certificate. The CA public key verifies all the certificates issued by the CA and is vital for the proper operation of the PKI.

- For many systems such as web browsers, the distribution of CA certificates is handled automatically.

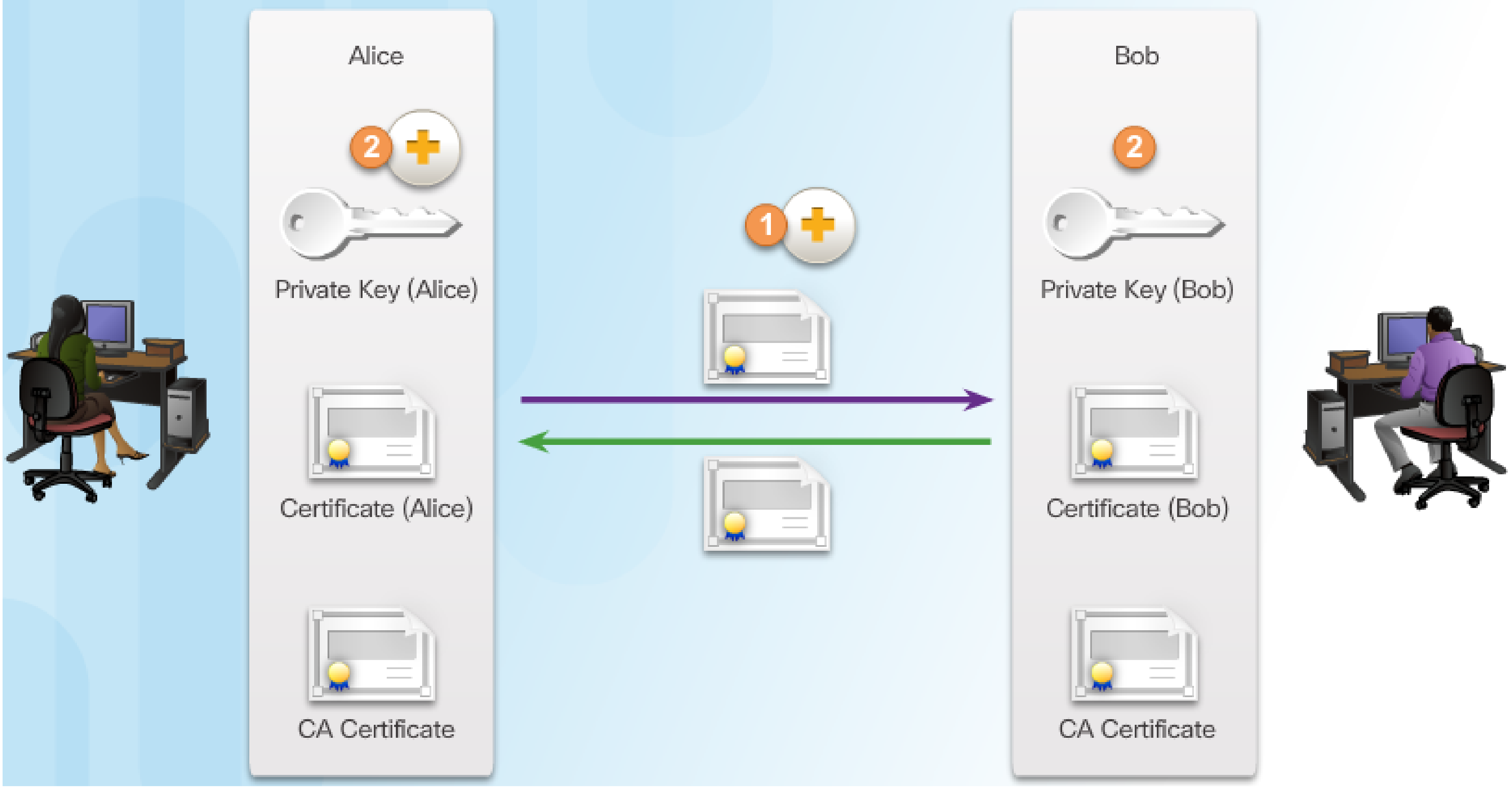
- The certificate enrollment process is used by a host system to enroll with a PKI. To do so, CA certificates are retrieved in-band over a network, and the authentication is done out-of-band (OOB) using the telephone.

- Authentication no longer requires the presence of the CA server, and each user exchanges their certificates containing public keys.

- Certificates must sometimes be revoked. The two of the most common methods of revocation are Certificate Revocation List (CRL) and Online Certificate Status Protocol (OCSP).

Digital Certificates and CAs - processes

- And they may start communicate securely



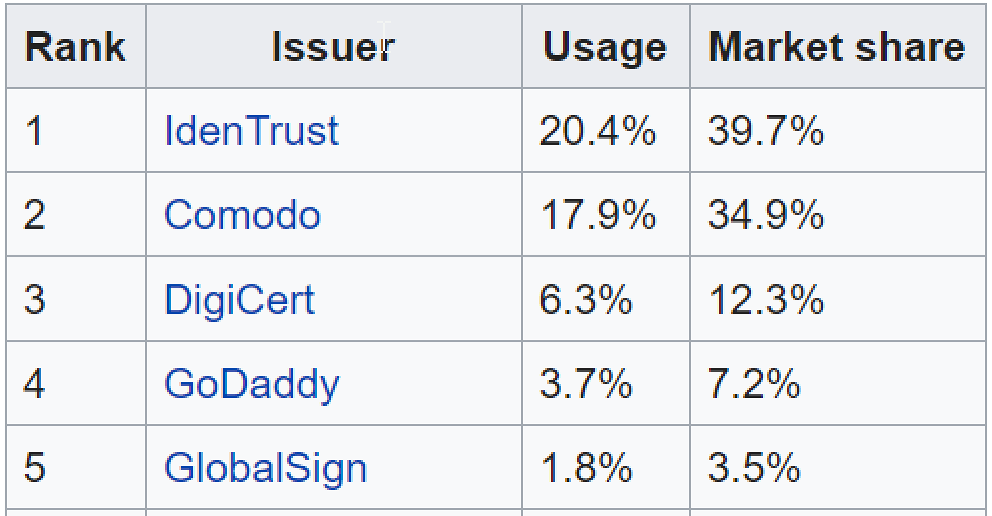
PKI Applications

- PKI in an enterprise?
  - SSL/TLS certificate-based peer authentication
  - Secure network traffic using IPsec VPNs
  - HTTPS Web traffic
  - Control access to the network using 802.1x authentication
  - Secure email using the S/MIME protocol
  - Secure instant messaging
  - Approve and authorize applications with Code Signing
  - Protect user data with the Encryption File System (EFS)
  - Implement two-factor authentication with smart cards
  - Securing USB storage devices

https://www.verisign.sk/?gclid=Cj0KCQjwzozsBRCNARIsAEM9kBOU_CXDk39Ke46vy5mEB-yUoU6Vt0zpTyw0kPk3V5_J2G11MvkWoooaAiMBEALw_wcB

Final notes

- PKI is mature technology, widely supported
- Problems
  - Complex and pricy to built up
  - How to built net of trust
    - Lot of local, regional, national or global PKI providers
  - PKI and certificate issuing is service provided usually for a price
    - Ignoring self-signed certs
    - Price depends
      - Type of cert (email, ssl/tls), validity, warranty: from tens to hundreds dollars per year and cert



Rank	Issuer	Usage	Market share
1	IdenTrust	20.4%	39.7%
2	Comodo	17.9%	34.9%
3	DigiCert	6.3%	12.3%
4	GoDaddy	3.7%	7.2%
5	GlobalSign	1.8%	3.5%

- Trends
  - Open CA – global SSL/TLS certs free of charge
    - Let's Encrypt
    - CAcert.org
  - SSL Free certs
  - BlockChain PKI
    - CertCoin
    - FlyClient
    - BlockQuick
- Critisim
  - Price
  - Security issues
    - CA compromise, attack on key storage, implementation weaknesses,

Vocabulary

- Advanced Encryption Standard (AES)
- Certificate Authority (CA)
- Certificate Revocation List (CRL)
- Data Encryption Standard (DES) and 3DES
- Diffie-Hellman (DH)
- Digital Signature Algorithm (DSA)
- Digital Signature Standard (DSS)
- hash-based message authentication code (HMAC)
- Internet Key Exchange (IKE)
- Lightweight Directory Access Protocol (LDAP)
- Public Key Infrastructure (PKI)
- Rivest ciphers (RC) series algorithms
- MD5 message-digest algorithm
- Pretty Good Privacy (PGP)
- Rivest, Shamir, and Adleman (RSA) algorithm
- Secure Hash Algorithms (SHA)
- Secure Socket Layer (SSL)
- Software-Optimized Encryption Algorithm (SEAL)
- Transport Layer Security (TLS)

UNIVERSITY OF ŽILINA
Faculty of Management Science
and Informatics

KATEDRA INFORMAČNÝCH SIETÍ * FAKULTA RIADENIA A INFORMATIKY ŽU *

CISCO
Networking
Academy

MINISTERSTVO
ŠKOLSTVA, VEDY,
VÝSKUMU A ŠPORTU
SLOVENSKEJ REPUBLIKY

KoNiEc